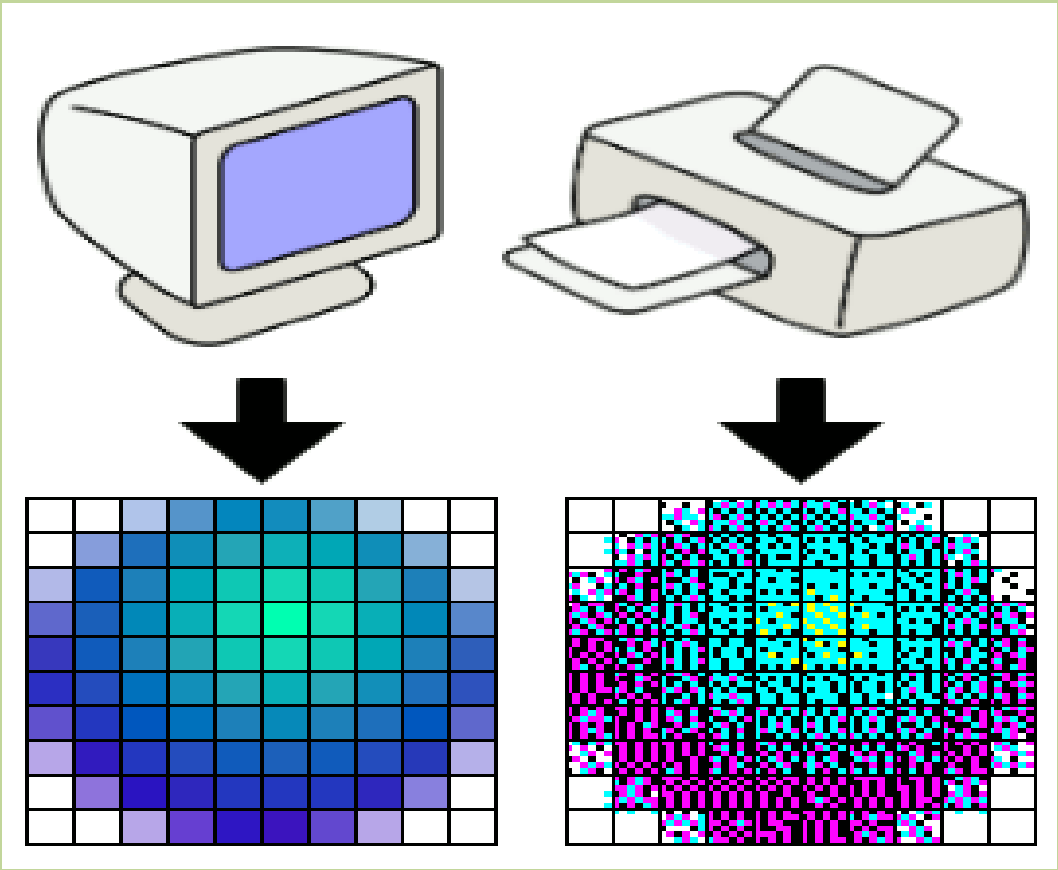
مقدمه‌ای بر پردازش تصویر با

MATLAB

(برای دانشجویان نرم افزار کامپیوتر و فناوری اطلاعات)



مولف:

مرضیه فریدی ماسوله

(مدرس دانشگاه آزاد اسلامی و دانشگاه جامع علمی کاربردی)

به نام خدا

منابع:

۱) پردازش تصویر دیجیتال نویسنده گونزالس

۲) پردازش تصویر در MATLAB نویسنده گونزالس

۳) MATLAB برنامه نویسی برای مهندسان نویسنده چاپمن

۴) آموزش کاربردی مهندسی برق با MATLAB نویسنده نیما جمشیدی

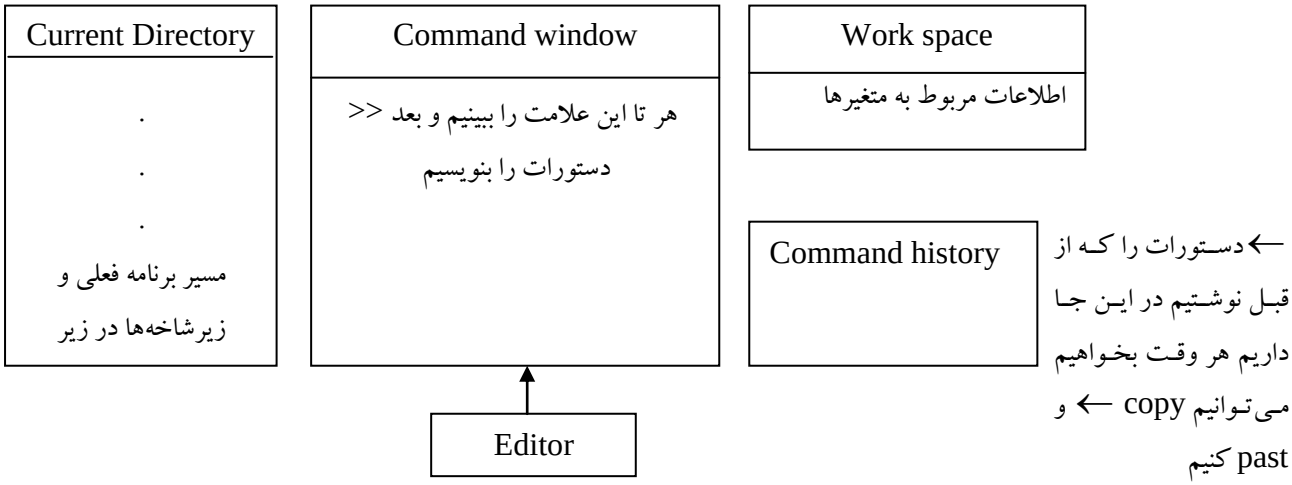
بخش اول

آشنایی ابتدایی با نرم افزار MATLAB

بخش اول

آشنایی ابتدایی با نرم افزار MATLAB

در صفحه کامپیوتر محیط MATLAB ما دارای چهار پنجره می باشد:



Work space: اطلاعات مربوط به متغیرها و داده‌های که برای اجرا در پنجره command window (یا اجرای دستورات) نوشته نمایش می‌دهد.

'2' ← یک کاراکتر ۲ یک مقدار

تعریف متغیر در MATLAB :

زمانی متغیر تعریف می شود که مقدار یا ارزشی به آن تعلق گیرد . در تعریف یک متغیر باید به نکات زیر توجه شود:

۱ -به حروف بزرگ و کوچک حساس است .

۲ -نام متغیر نمی تواند با یک عدد شروع شود .

۳ -باید دقت کرد که نام متغیر از کلمات تعریف شده در خود نرم افزار نباشد.

۴ -از _ (underline) در تعریف نام متغیر می توان استفاده کرد.

۵ -طول متغیر بین ۱ تا ۳۲ کاراکتر باشد،

۶ -اسم متغیر با اسم کلید اصلی (نام برنامه) نباید یکی باشد.

در command windows اگر بنویسیم is keyword>> لیست همه کلمات کلیدی را به ما میدهند در MATLAB زمانی متغیرها تعریف می‌شوند که به آن‌ها مقدار نسبت داده شود وگرنه مقدار نداشته باشد خطا می‌دهد.

دستور انتساب در MATLAB:

>>A=2 ⟸ نوشته می‌شود در ←

Work space	
A	**2**

اگر بنویسیم A=2 و در انتهای خط

دستور سمیکلون نگذاریم در خط بعد نمایش می دهد:

Ans

A=2 را دوباره نمایش می‌دهد اگر نخواهیم دوباره پائین نوشته شود; می گذاریم.

>>A=2;

>>B=A+2

Work space	
A	**2**
B	**A+2=4**

علامت ٪ برای توضیحات هر خط دستور می‌باشد.

مثلاً >>A=2% A is a variable

دستور خروج از MATLAB >>exit

>>quit

انواع دستورات در MATLAB:

فرمان: بعد نام فرمان نوشته و وردی جلوش نوشته می‌شود.

تابعی: اما دستورات تابعی ورودی در داخل () پرانتز نوشته می‌شود.

دستور برای پاک کردن صفحه می‌باشد >>CLC از صفحه command window نوشته‌ها پاک

دستور DIR: اسامی زیر ← Directori و فایل‌های یک Directori را لیست می‌کند.

>>dir اسم فایل % توضیحات

>> dir MATLAB1 دستور فرمانی

مثال

>> dir (MATLAB) دستور تابعی

Mk (میک) برای ساخت یک دایرکتوری

>> mkdir MATLAB2 % command mode

مثال

>> mkdir ('MATLAB2')% functional mode

برای حذف فایل

>>delet filename فرمانی

>>delet ('filename') تابع

توضیح: بعد ٪ حالت فانکشنال یا کامند یا دستوری

اگر بخواهیم فقط متغیر حذف شود از clear

برای حذف متغیرها از clear

>>clear A

برای فعال کردن (باز کردن work space) در صورتی که فعال نباشد، معمولاً فعال

>>work space

دستور who تمام متغیرهای تعریف شده را لیست می‌کند. به ما نمایش می‌دهد.

>>who

دستور whos علاوه بر اسم متغیر،سایز، بایت و کلاسیش را هم نمایش می‌دهد.

>>whos

دستور what فایل‌های MATLAB را که در فضای کاری موجود هست لیست می‌کند.

>>what

دستور which اسم متغیر را جلوش می‌یاریم اگر قبلاً با این اسم داشتیم به ما پیغام می‌دهد.

جدول الویت برای عملگرهای حسابی

() ← پرانیز

توان ^

مثال:>>2*(3+4/2)

* / \

ضرب و تقسیم

جمع و تفریق

نکته اگر علامت تقسیم بر عکس بود: مانند 2\4 ۲ را تقسیم بر ۴ می‌کند تقسیم راست

عملگرهای منطقی.

And (c1,c2) ← c1 & c2

Not (c1) ← -c1

OR (c1,c2) ← c1 or c2

به فرض ما یک آرایه سطری داریم: برای دسترسی به عنصر کافیست دستور زیرا را بنویسم

سطری>> R=[2 5 4 8] $\overset{\text{دستور}}{\rightarrow}$ >>R (1,3) ans 4

در محیط MATLAB با

$$R۲ = \begin{bmatrix} a \\ b \\ d \\ e \end{bmatrix} \rightarrow \qquad >> R_۲(۳,۱)\hookleftarrow \qquad ans \qquad d$$

در محیط MATLAB ما که نمی‌توانیم ماتریس بکشیم به صورت زیر نمایش می‌دهیم.

>>R=[1,2,3,4] سطری

>>R_2=[1,2,3,4]

ایجاد آرایه منظم:

آرایه‌ای دارد از ۱ تا ۱۵ برای ایجادش کافیست اسم آرایه نوشته شود (این جا R3).

اگر انتهای دستور ; بگذاریم هیچی نشون نمی‌دهد می‌رود دستور بعد >>R3=1:1:15

اما اگر نذاریم ↵بزنیم همان لحظه چیزی را که ایجاد کردیم فوراً نشان میدهد. ←

R3=[1,2,3,…15]

>>RB = -1:-1:-15 → برای منفی

آرایه داریم و می‌خواهیم تعداد عناصر آرایه را نمایش بدهیم:

داریم >>X=[1,2,3]

فرض آرایه‌ای مثل X داریم می‌خواهیم سه تا دیکه هم اضافه می‌یایم خط فرمان بعدی

↵ >>X=[X,4,5,6] اضافه می کنیم

Anser

X=

[1,2,3,4,5,6]

یه زمانی می‌خواهیم به ابتدای آرایه عنصر اضافه شود

>>x=[6,7,8,x]

X=[6 7 8 1 2 3]

الحاق دو آرایه:

>>Z=[x,y]

تابع fliper: معکوس کردن عناصر آرایه

تابع sort: به صورت نزولی آرایه را مرتب می‌کند.

برای صعودی- می‌توانیم اول به صورت نزولی مرتب بعد با flipper عکس کنیم.

جمع آرایه با یک مقدار:

با یک مقدار ضرب می‌شود و یا با یک مقدار جمع $\rightarrow$B(i)=A(i)+2

جمع دو آرایه در صورتی که طول و نوعشان یکسان باشد، برای تفریق هم این مسئله صدق می‌کند.

اگر آرایه یک عنصر کم داره و باید حتماً عمل جمع یا تفریق انجام داد به جای عنصر-۰- در نظر گرفته می‌شود.

مثلاً $\begin{cases} y_1 = x^3 + 2x^2 + 1 \Rightarrow x^3 \quad 2x^2 \quad 0x1 \\ y_2 = x + 5 \quad\quad\quad \Rightarrow 0 \times x^3 + 0 \times x^2 + x + 5 \end{cases} \Rightarrow x^3 + 2x^2 + x + 6$

$$\begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 6 \end{bmatrix}$$

ضرب عنصر به عنصر: برای ضرب عنصر به عنصر آرایه از علامت **.*** استفاده می‌شود.

مثال: A **.*** B

مثلا $A.*B = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 5 \end{bmatrix}$

توابع floor و ceil:

>>a=floor (5.7) ↵

a=5

Floor: بزرگترین عدد صحیح کوچکتر یا ماسوی x را میدهد.

Ceil: کوچکترین عدد صحیح بزرگتر یا مساوی x

>>ceil (8.2) ↵

بزرگترین مقسوم علیه مشترک g c d

کوچکترین مضرب مشترک L c m

مثال:>>gcd(78,42) ↵

>> lcm (8,2) ↵

Plaot: برای رسم نمودارها به کار می‌رود.

Y=sin x

$Y=x^2$

Plot(x,sin(x))

>>x=0:0.1:10;

>>plot (x,y) ↵

وقتی تابعی که تمام اعداد را بگیرد جلوش صفر می‌گذاریم که عدد را به هر توان می‌رساند می‌خواهیم چند تا نقطه را به توان برسانیم (فقط یک عدد نه) جلوش نقطه می‌گذاریم.

>>x=[1 2 3]

>> $y=x^2$

مثال : $y=x^2$

Plot ($x,x^2.$)

>>x=0:05:20;

>>$y=x^2.$;

>>plot (x,x^2)

دستور LOOK FOR: برای جستجوی یک رشته در MATLAB تمام رشته‌هایی که sin یا nsin یا sin.cos و ... در کل خود عبارت یا عین عبارت را نشان می‌دهد.

مثال look for sin

مثلاً درسمان را بدیم در نرم افزار نیست مگر اینکه از قبل اسمی را به نامی که می خواهیم از قبل ذخیره کرده و داشته باشیم:

تمام رشته‌های مشابه را هم نام میکند تولید توانها در :

>> w=[1,2,3] 1 2 3

W=[1,2,3] 1 2 3

>>v=[1.2.3]

مثال>> v' ⇒ 1 2 3 سطری

>> w' ⇒ 1

2

3

مثال >> [1 2 3]'

1

2

3

>> [1,2,3]'

1 2 3

ضرب داخلی:

$$\mathbf{[a1 \quad a2 \quad a3]}\begin{bmatrix}\mathbf{b1}\\ \mathbf{b2}\\ \mathbf{b3}\end{bmatrix} = a_1b_1 + a_2b_2 + a_3b_3$$

>> ***A* B***

>> ***A*[1 2 3];

>> ***B*** = [1 2 3];

>> 1 + 4 + 9 = 14

روش های معرفی ماتریس:

ایجاد یک ماتریس با M سطر و N ستون:

اعضای ماتریس باید در داخل براکت [] تعریف شود.

ایجاد ماتریس به روش سطری:

اگر بخواهیم یک ماتریس سطری در نرم افزار MATLAB ایجاد کنیم:

ستون

```
>> a=[1,2,3,4]
Or
>> a=[1 2 3 4]
```

خروجی

[1 2 3 4]

سطر

```
>>a=[1 2 3;4 5 6;7 8 9]
```

خروجی

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

دستور:

>> A=عدد شروع : گام یا پرش : عدد پایانی

مثال:

ماتریسی تولید کنید که اعداد زوج بین ۱ تا ۱۰۰ را نشان دهد.

```
>> A=2:2:100
```

خروجی

2 4 6 8 100

نکته : وجود ; در انتهای هر خط فرمان مانع از اجرای خروجی در همان لحظه می شود.

مثال:

A=2:2:100
B=1:2:100

خروجی

$$\begin{pmatrix} 2 & 4 & 6 & 8 & \dots \\ & & & & 100 \\ 1 & 3 & 5 & 7 & \dots. \end{pmatrix}$$

C=[A;B]

دستیابی به آرایه ای مشخص از یک ماتریس:

1) **A=B(x,y)**

از ماتریس B سطر X و ستون Y جدا شده ، در یک متغیر مانند A قرار داده می شود.

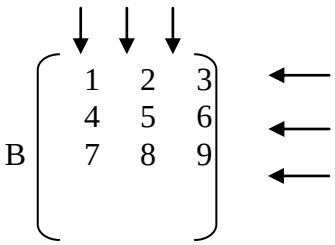
2) **A=B(x,:)**

از ماتریس B به ازای تمام ستون ها ،سطر X جدا شده ، در متغیر A قرار می گیرد.

3) **A=B(:,y)**

از ماتریس B به ازای تمام سطرها ، ستون Y جدا شده و در متغیری مانند A قرار می گیرد.

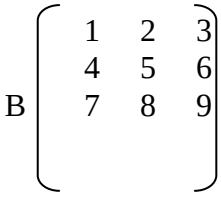
مثال:

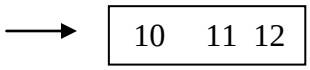


A=B(2,:) ⟶ A = [4 5 6]

C=B(:,3) ⟶ C = 3 6 9

اضافه کردن یک سطر جدید به ماتریس:





B=[B;[سطر مورد نظر]]

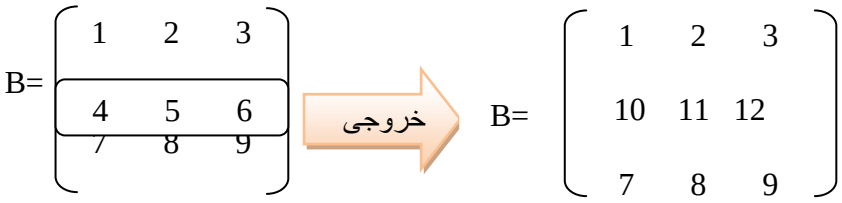
>>B=[B;[10,11,12]]

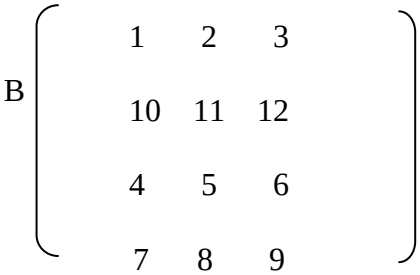
ترانهاده:

جای سطر و ستون را عوض میکند.

B=[2,:]=[10 11 12]

B=[2,:]=[10 11 12]





اضافه کردن یک ستون جدید :

>>B = [B [10 ; 11 ; 12]]

>>B = [B [10 , 11 , 12]]

```
B [ 4 , : ] = [ 7  8  9 ]
B [ 3 , : ] = [ 4  5  6 ]
B [2 , : ] = [ 10  11  12 ]
```

اضافه کردن یک ستون جدید :

>>B = [B [10 ; 11 ; 12]]
>>B = [B [10 , 11 , 12]]

ایجاد ماتریس با استفاده از توابع:

یک ماتریس واحد ایجاد می کند ⟵ >>A=eye[تعداد ستون , تعداد سطر]

ماتریس واحد:

ماتریسی است که قطر اصلی یک (۱) ، و سایر درایه ها صفر (۰) باشد.

مثال:

eye(3,3) خروجی

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

ماتریس دو بعدی:

```
>> A=[5,6,7;8,9,10;11,12,13]
     5   6   7
```

8 9 10

11 12 13

دترمینان:

مثال A=[5,6,7,8,9,10,11,12,13]'

5 8 11

6 9 12

7 10 13

مثال >> A=[1,2,3,4,5,6,7,8,9]

>> B=[1,2,3,4,5,6,7,8,9]

ضرب در MATLAB >>A*B

توابع از پیش تعریف شده:

تابع ones :

یک ماتریس ایجاد می کند که تمام عناصرش یک (۱) می باشد.

>> A=ones(تعداد ستون , تعداد سطر)

Example:

Ones(2,3) خروجی

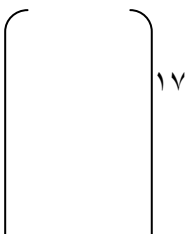
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

تابع zeros :

یک ماتریس ایجاد می کند که تمام عناصرش صفر (۰) است.

>>A=zeros(تعداد ستون , تعداد سطر)

Example:



Zeros(4,2)

0 0
0 0
0 0
0 0

>>A=Rand(تعداد ستون , تعداد سطر)

اعداد تصادفی بین صفر و یک تولید می کند.این تابع برای ایجاد noise در سیستم به کار می رود.اعداد مثبت ، تصادفی و تا ۴ رقم اعشار هستند.

>>A=magic(ستون,عدد سطر)

تابع magic یک ماتریس مربعی تصادفی تولید می کند که جمع سطر و ستون های آن یکسان است .

>>A=unique(B)

این تابع تمام درایه های ماتریس B را به صورت صعودی و غیر تکراری دسته بندی کرده و در متغیر جدیدی به نام A قرار میدهد.

Example:

B = [2 1 1 1 3 3 10 4 4]
A = [1 2 3 4 10]

توابع اطلاعاتی ماتریس:

اولین تابع Size می باشد.این تابع اندازه ماتریس را به صورت ۲ عدد که اولین آن تعداد سطر و دومین آن تعداد ستون ماتریس است نمایش می دهد.

Size (اسم ماتریس)

Example:

C $\begin{bmatrix} 1 & 2 \\ & 3 \\ 4 & 5 \end{bmatrix}$

>> D = Size (C)

>> A = length (B)

این دستور تعداد آرایه های ماتریس B را در متغیری مانند A قرار می دهد.

متغیر clear ⟵ متغیر حذف

متغیر clc ⟵ صفحه فرمان را پاک می کند

حاصل این تابع یک ماتریس سطری است که آرایه های موجود در ستون ماتریس B را جمع کرده و حاصل را در متغیری مانند A قرار می دهد.

$$\begin{pmatrix} 10 & 20 & 30 \\ 11 & 2 & 3 \\ 12 & 22 & 33 \end{pmatrix}$$

سطر[66 44 33]

>>plot(x)

توسط این دستور می توانیم ماتریس های خطی ، بردارها و حتی توابع را رسم کنید.

X = [10 20 30]

```
>>plot(x)
>>y=[1,2,3]
>>z=(x,y)
>>plot(2)
```

>>A=max(B)

حاصل آن یک ماتریس سطری است که هر عنصر آن ماکسیمم ستون مورد نظر در ماتریس B است.

>>A=min(B)

حاصل آن یک ماتریس سطری است که هر عنصر آن مینیمم ستون مورد نظر در ماتریس B است.

>>A=mean(B)

حاصل آن یک ماتریس سطری است که عناصر ان میانگین ستون مورد نظر است.(هر عنصر آن میانگین ستون مورد نظر است)

مرتب سازی ستون ماتریس:

این تابع آرایه های ستونهای ماتریس را به صورت صعودی مرتب کرده و در متغیر جدید قرار می دهد.

$$B = \begin{pmatrix} 5 & 6 & 7 \\ 3 & 2 & 1 \\ 1 & 3 & 5 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 2 & 1 \\ 3 & 3 & 5 \\ 5 & 6 & 7 \end{pmatrix}$$

تابع فاکتوریل:

این تابع به ازای تک تک عناصر ماتریس فاکتوریل را محاسبه می کند.

A=factorial(B)

$$A = \begin{pmatrix} 7! & 6! & 5! \\ 2! & 1! & 3! \\ 5! & 3! & 1! \end{pmatrix}$$

تابع زیر ماتریس را به اندازه ۹۰ درجه در جهت خلاف عقربه های ساعت می چرخاند.

A=rot90(B)

$$\begin{pmatrix} 7 & 1 & 5 \\ 6 & 2 & 3 \\ 5 & 3 & 1 \end{pmatrix}$$

دستورات حلقه و شرط:

For نام متغیر = نقطه شروع : گام : نقطه پایان
عملیات مورد نظر
End

While (شرط)

عملیات مورد نظر
End

شرط اول If
عملیات
Else
عملیات
End

شرط اول If
عملیات
شرط دوم Else if
عملیات
…
End

مثال:

برنامه ای بنویسید که اعداد زوج بین ۱ تا ۱۰۰ را جمع کرده و در متغیر A و B بریزد.

```
A=0
For C=2:2:100;
A=A+C
End
B=A
```

Eye: تابع eye تابعی است که ماتریس تولید می‌کند که قطر اصلی او بقیه صفر می‌باشد.

مثال A=eye (4)

Diag: تابع diag تولید یک ماتریس قطری با اعداد خاص روی قطر و بقیه صفر

مثال A=[1 2 3];

```
>> diag (A)
```

کار دیگر diag مثلاً ما یه ماتریس به نام A داریم قطر ماتریس را به ما برمی‌گرداند.

>>A=[1 2, 3 4]

$\Rightarrow$>> diag (A)

قطر ماتریس را برمی‌گرداند که ۴ و ۱

تابع ones تابعی است که n*n که تمام عناصر آن یک است.

مثال >> ones (4)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

تابع zeros تابع n*n که تمام عناصر آن صفر است.

>>zeros (3)

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

تابع Rand: تابعی n*m که اعداد آن تصادفی و بین صفر و یک است.

>>Rand(3) $\begin{bmatrix} 3*3 \text{ تابع} \\ \text{با اعداد ۰ و ۱ یا} \\ \text{اعشاری} \end{bmatrix}$ فعل >> **Rand**(2,3) $\begin{bmatrix} x & y & z \\ x' & y' & z' \end{bmatrix}$

>>D=Rand (2,1) * 10 $\begin{bmatrix} x \\ y \end{bmatrix}$ ۱۰ هم ۲ و هم ۱ در ۱۰ ضرب می‌شود.

Round: تابعی که اعداد با مقدار اعشاری بزرگتر از ۰,۵ را به ۱ و کوچکتر از ۰,۵ را به صفر رند می کند.

بخش دوم

مقدمه ی بر پردازش تصویر

بخش دوم

مقدمه ی پردازش تصویر:

چشم به عنوان یکی از حس‌گرهای انسانی نقش بسزایی در زندگی دارد. امروزه با پیشرفت چشمگیری که در ساخت پردازنده‌های صورت گرفته است این امکان فراهم نشده تا در ساعت زبانها و سیستم‌های کنترلی از دوربین به عنوان یک چشم مصنوعی استفاده کنیم.

کاربردهای عمده پردازش تصویر:

۱-رباتیک ۲-سیستم‌های دفاعی ۳-مهندسی پزشکی ۴-کنترل صنعتی ۵-گرافیک کامپیوتر

۶-notion rtachiy «ردیابی حرکت» ۷-تشخیص پلاک خودرو

در سیستم‌های رباتیک معمولاً از پردازش تصویر برای هدایت ربات و تشخیص اشیاء استفاده می‌شود. در سیستم‌های دفاعی برای یافتن هدف و یا رهگیری یک هدف متحرک پردازش تصویر یکی از قابل اعتمادترین روشهای موجود است.

در مورد کاربردهای گرافیکی یکی از معروفترین نرم‌افزارهای مبتنی بر پردازش تصویر photoshop و پس از آن یک نرم افزار محاسباتی دقیق به نام MATLAB وجود دارد.

برای مهندسی پزشکی از رایج ترین کاربردهای پردازش در پزشکی

خط تولید صنعتی: برای کنترل کیفیت محصولات تولید شده و همچنین کنترل حرکات خط تولید از سیستم‌های مبتنی بر پردازش تصویر استفاده می‌کنند.

یک تصویر از لحظه ورود به سیستم تا تولید تصویر خروجی به ترتیب مراحل زیر را طی می‌کند.

۱-دریافت تصویر ورودی: در این مرحله تصویر از ورودی خوانده شده وارد سیستم می‌گردد، تصویر ورودی می‌تواند بر روی ابزار ذخیره‌سازی قرار گرفته و یا از یک دوربین گرفته شود و به عنوان مثال در سیستم تشخیص اثرانگشت، تصویر ورودی از طریق scanner اثرانگشت را وارد سیستم می‌کند.

۲-پیش پردازش تصویر: هدف تولید در این مرحله ارتفاع بهبود کیفیت تصویر و حذف مولفه‌های غیرضروری از سیستم است. به عنوان مثال: خشکی پوست، جراحت، و یا عدم تمیزی از مواردی هستند که قبل از پردزاش تصویر باید با استفاده از تکنیک‌های پردازش تصویر به رفع آنها پرداخت.

۳-پردازش تصویر: هدف کلی این مرحله، شناسایی ویژگی‌هایی از تصویر است که بتوان از آنها برای کاربرد موردنظر استفاده کرد، شناسایی نقاط ویژه، مرکز ثقل و انحنا از جمله ویژگیهایی هستند که در سیستم تشخیص اثرانگشت می‌توان استخراج کرد.

۴-آنالیز تصویر: در این مرحله با استفاده از ویژگی‌های استخراج شده به آنالیز تصویر می‌پردازیم، به عنوان مثال: پس از شناسایی نقاط ویژه و انحنا در سیستم اثرانگشت با آنالیز تصویر سعی می‌کنیم. شخص متناظر با اثرانگشت را شناسایی می‌کنیم، در آنالیز تصویر معمولاًاز تکنیک‌های هوش مصنوعی مانند شبکه‌های عصبی، درخت تصمیم و کلاس‌بندی استفاده می‌کنیم.

تصویر دیجیتال: یک تصویر را می‌توان توسط تابع دو بعدی $f(x,y)$ نشان داد که در آن x,y مختصات مکانی و f درهر نقطه شدت روشنایی تصویر در آن نقطه را نشان می‌دهد.

اصطلاح سطح خاکستری و یا grayscal یا graylevel به شدت روشنایی سطح تصویر اطلاق می-شود. تصاویر رنگی از تعدادی تصویر دو بعدی تشکیل شده‌اند.

زمانی که مقادیر x,y و مقدار $f(x,y)$، مقداری گسسته و نامحدود باشد تصویر را دیجیتالی می‌نامند دیجیتالی کردن مقادیر x,y را sampling می‌گویند.

برای نمایش یک تصویر m*n از یک آرایه دو بعدی یا ماتریس که در m سطر و n ستون ذخیره می-شوند استفاده می‌کنیم، مقدار هر عنصر از آرایه نشان دهنده شدت روشنایی تصویر در آن نقطه است هر عنصر آرایه یک مقدار ۸ بیتی است. که می‌تواند بین ۰ تا ۲۵۵ باشد.

مقدار صفر نشان دهنده رنگ تیره یا سیاه و مقدار ۲۵۵ نشان دهنده رنگ روشن یا سفید است.

ردیابی حرکت: مثلاً زمانی که دو وسیله نقلیه در حال حرکت هستند و ما می‌خواهیم آنها را در حال حرکت تحت نظر قرار دهیم می‌توانیم با مشخص کردن یک ویژگی از تصاویر، که این ویژگی تغییرناپذیر باشد از پردازش تصویر استفاده کنیم. منظور از تغییر ناپذیربودن این است که :

۱-تغییرناپذیر با انتقال

۲-تغییرناپذیر با مقیاس

۳-تغییرناپذیر با چرخش

۳-تشخیص بلوک خودرو از طریق پردازش تصویر و شبکه عصبی.

مرحله اول: عکس توسط دوربین گرفته می‌شود.

مرحله دوم: از عکس دریافت شده مشتق گرفته می‌شود (برای تعیین لبه تصویر بکار می‌رود).

مرحله سوم: نقاط توخالی پر می‌شود.

مرحله چهارم: از میان تمام عناصر فقط عنصر با شرایط خاص انتخاب می‌شود.

مرحله هفتم: عکس رنگی معادل عنصر انتخاب شده، جدا می‌شود.

مرحله ششم: عکس سیاه و سفید به کد ۱ و صفر یا باینری تبدیل می‌شود.

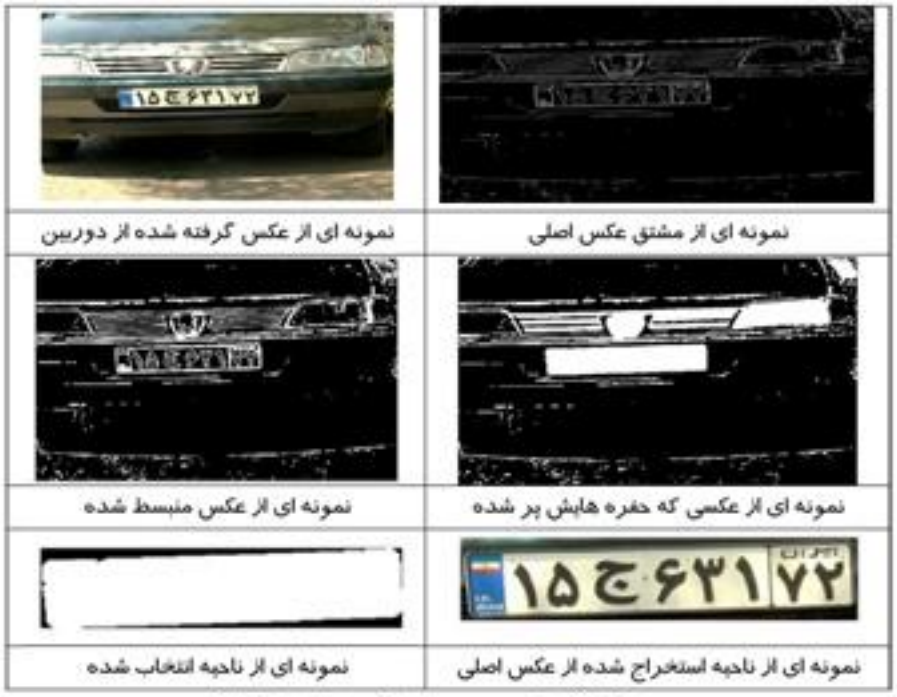
مرحله هشتم: عکس سیاه و سفید در راستای محور افقی تنظیم می‌شود.

مرحله نهم: عکس سیاه و سفید را از هر گونه nois پاک می‌کنیم.

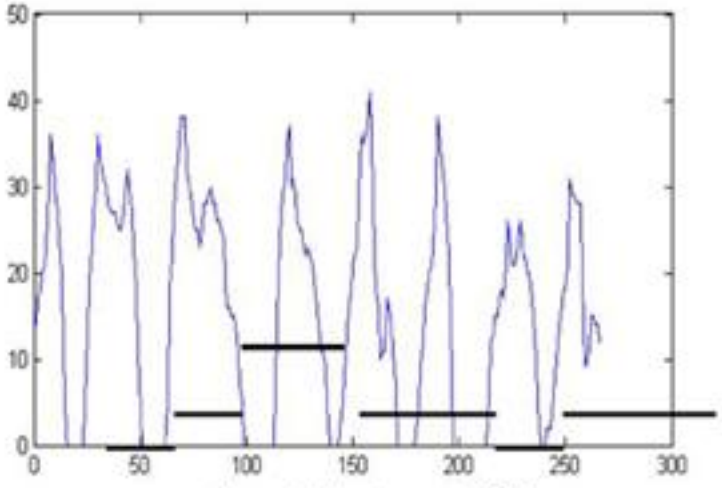
مرحله یازدهم: در نهایت اعداد پلاک بدست می‌آید.

مرحله دوازدهم: عکس سیاه و سفید حاصل پلاک توسط الگوریتمی اعدادش از هم تفکیک شده و هر عدد به طور جداگانه به شبکه عصبی وارد می‌شود تا عدد موردنظر را شناسایی کند.

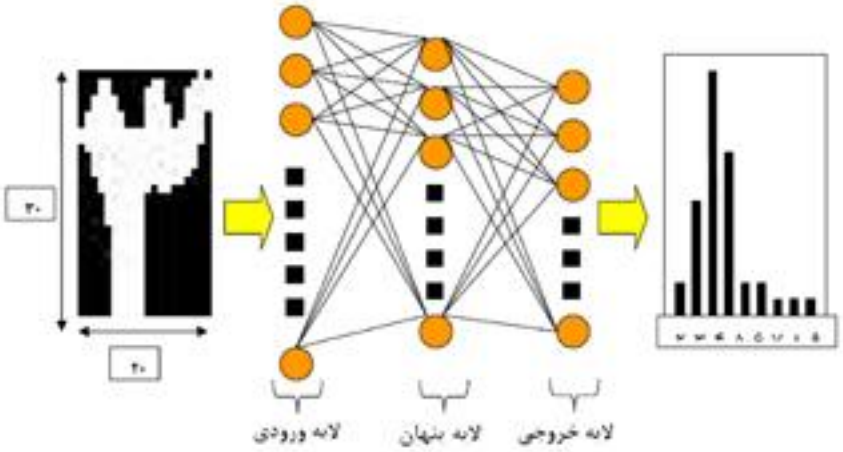
شکل



شکل ۳ . تصاویر مربوط به الگوریتم استخراج پلاک



شکل ٤ . تصویر ستون های تصویر باینری



شکل ۵. تصویر الگوریتم تشخیص اعداد و شبکه عصبی

تاریخچه پردازش تصویر:

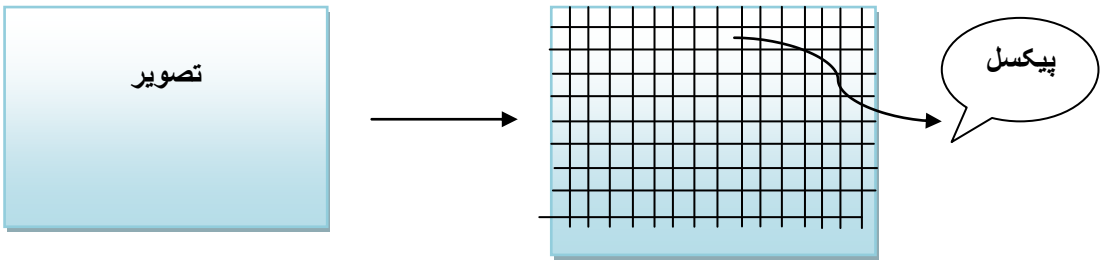
سالهای قبل از ۱۹۲۰ اولین کاربرد پردازش تصویر در صنعت روزنامه بود، با استفاده از یک متد، تصاویر کد شده و از طریق کابل زیردریایی به مکان دیگری ارسال شده و در محل دریافت دوباره به حالت اول برمی‌گشت از اواسط تا آخر دهه ۱۹۲۰، افزایش بهبود کیفیت تصویر، ایجاد فرآیندهای جدید براساس تکنیک‌های عکاسی، افزایش پیکسل‌های تصویر از پیشرفت‌های این دوره بوده (دهه ۱۹۶۰) با بهبود تکنولوژی کامپیوتر رقابت در پردازش تصویر آغاز شده و اساس پردازش تصویر بر پایه کامپیوتر بنا نهاده شده. دهه ۱۹۷۰ پردازش تصویر در علوم پزشکی مورد استفاده قرار گرفت. دهه ۱۹۸۰ تاکنون استفاده از تکنیک‌های پردازش تصویر گسترش یافته و امروزه در هر کار و محیطی مورد استفاده قرار می‌گیرد.

نکته: یکی از رایج‌ترین کاربردهای پردازش تصویر بهبود کیفیت تصویر و حذف noise از تصویر است.

مفاهیم رنگ و تصویر:

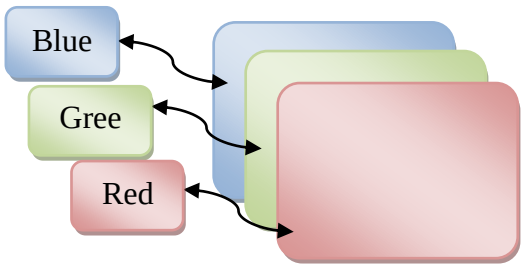
پیکسل:

تصاویری که توسط یک دوربین دیجیتال دریافت می شود از نقاط مربع شکل کوچک که هر کدام رنگ خاصی دارند تشکیل شده است،که به این نقاط پیکسل می گویند و کوچکترین عنصر تشکیل دهنده یک تصویر است .هر پیکسل باید به صورت یک عدد بیان شود تا قابل پردازش باشد.



مدل رنگ RGB :

از سه رنگ قرمز ، سبز ، و آبی تشکیل شده که همه رنگ ها از ترکیب این سه رنگ حاصل می شوند.



شناسایی تصویر در MATLAB :

برای پردازش و تحلیل (آنالیز) تصاویر در نرم افزار MATLAB ابتدا تصویر باید برای سیستم قابل فهم شود و این کار تنها با تبدیل تصاویر به ماتریس ها امکان پذیر است . نرم افزار MATLAB چهار نمونه از این تصاویر را تولید و بررسی می کند:

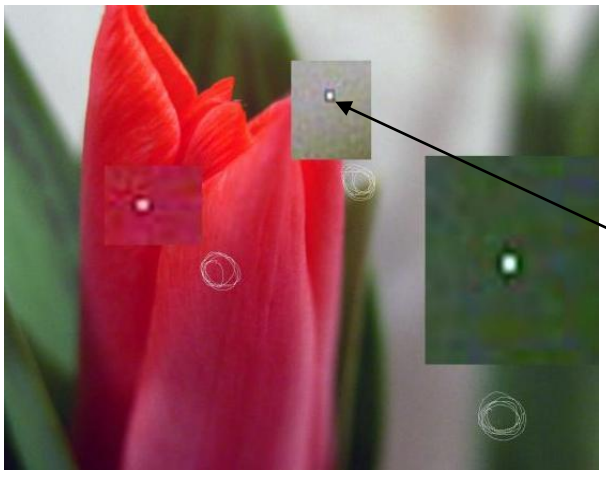
۱ -**تصاویر باینری:** ماتریس این تصاویر تنها از اعداد صفر و یک تشکیل شده است .(صفر نشان دهنده رنگ سیاه و یک نشان دهنده رنگ سفید است)

۲ -**تصاویر رنگی با شاخص (indexed):** تصاویر رنگی بعد از ورود به محیط MATLAB تبدیل به یک ماتریس می شود.در این نوع تصاویر اعداد ماتریس مشخص کننده ی رنگ پیکسل نیست بلکه شاخص یا عددی است که به یک جعبه ی رنگ اشاره دارد.این جعبه ی رنگ به صورت ۳ ماتریس برای رنگ های قرمز ، سبز و آبی در حافظه ی نرم افزار تعبیه شده است.با فراخوانی تصویر index شده ی جعبه ی رنگ نیز فراخوانی می شود.

۳ -**تصاویر غیر رنگی با شدت نور-تصاویر خاکستری (Gray):** این نوع تصاویر غیر رنگی بوده و روشنی و تیرگی تصویر را توسط یک ماتریس نمایش می دهد.آرایه های این ماتریس اعدادی بین صفر و یک با ۳ رقم اعشار است.در حال حاضر این تصاویر را می توان به صورت ۸ بیتی و مانند اعداد ماتریس های RGB نمایش داد با این تفاوت که در این نوع ماتریس ها رنگ وجود ندارد.

۴ -**تصاویر RGB :** این تصاویر از ۳ ماتریس تشکیل شده و هر پیکسل از ترکیب رنگ ۳ نقطه در ۳ ماتریس ایجاد می شود.

نمایش یک تصویر دیجیتال:



پیکسلی از تصویر

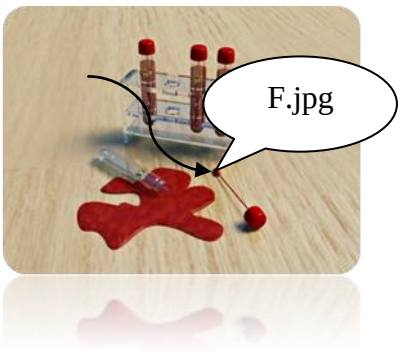
پر ← بازده مورد نظر ≥ مساحت سطح

خالی ← بازده مورد نظر ≤ مساحت سطح

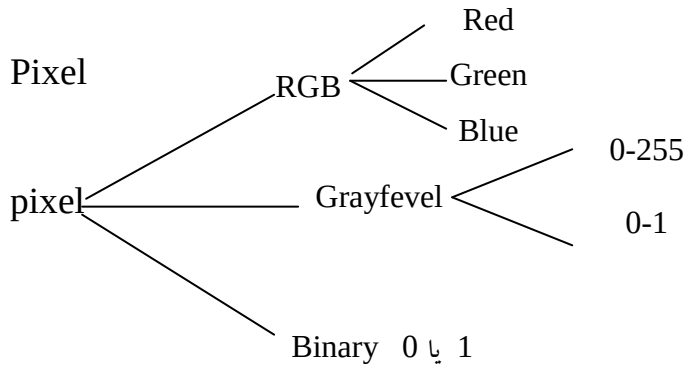
پردازش اولیه تصویر:

همانطور که گفته شد برای انجام هر نوع پردازشی بر روی یک تصویر ابتدا باید تصویر به یک ماتریس تبدیل شود.برای این کار از دستور زیر استفاده می شود.

A=imread(‘نام و پسوند تصویر’ ,مسیر)



عمل نرمالایز: در سطح خاکستری بازه پیکسل‌های مابین ۲۵۵ تا صفر است و هر پیکسل ۸ بیت را اشغال می‌کند برای نرمالایز کردن تصویر و تبدیل آن به بازه ۱ تا صفر کافیست تک تک پیکسل‌ها را بر ۲۵۵ تقسیم کنیم که باعث کاهش حجم حافظه مصرفی می‌شود.



*هر چه تقسیم‌بندی پیکسل‌های تصویر ریزتر باشد دقت تصویر مضاعف است.

در پردازش تصویر ۳ سطح داریم:

۱-سطح پائین: در سطح پائین، ورودی یک تصویر عملیات انجام شده بر روی آن، عملیات بهبود مانند حذف nois و خروجی آن یک تصویر است.

۲-سطح میانی: در سطح میانی ورودی یک تصویر عملیات روی آن شناسایی یک ویژگی و خروجی آن، ویژگی تصویر است.

۳-سطح بالا: ورودی یک ویژگی از تصویر مانند مساحت است و خروجی آن یک عمل یا عکس-العمل نسبت به آن ویژگی است مانند حرکت ربات به راست و چژ

	ورودی	عملیات	خروجی
سطح پائین	تصویر	حذف نویز	تصویر
سطح میانی	تصویر	شناسایی ویژگی	ویژگی تصویر
سطح بالا	ویژگی (مساحت)		عمل یا عکس العمل نسبت به ویژگی

پردازش تصویر دیجیتال بر دو عمل اصلی تأکید دارد.

۱-بهبود اطلاعات تصویر برای تغییر توسط انسان:

۲-پردازش داده های تصویر برای ذخیره سازی انتقال و نمایش مجدد آن برای درک توسط ماشین

نحوه نمایش ماتریس‌های تصویر

در سطح خاکستری تصویر تنها با یک ماتریس دو بعدی نمایش داده می‌شود که به ازای هر پیکسل یک عنصر در ماتریس وجود دارد در تصاویر RGB هر پیکسل با سه عنصر در سه ماتریس نمایش داده می‌شود. یعنی به ازای هر تصویر RGB سه ماتریس دو بعدی داریم که نماینده سه رنگ اصلی می‌باشد در تصاویر باینری یا (سیاه و سفید) هر پیکسل با مقدار صفر یا یک در ماتریس نشان داده می-شود در تصاویر index شده به جای استفاده از سه ماتریس، سه عدد نماینده هر پیکسل، index شده و تنها با همان عدد index شده نمایش داده می‌شود.

دستور Imread:

دستور imread تصویر مورد نظر را فراخوانی کرده و آن را در متغیری ریخته و تصویر را به شکل ماتریس در می‌آورد.

I=imread('c:\my computer\test.jpg')

مسیری که تصویر مورد نظر در آن قرار گرفته

نکته مهم در پردازش تصویر:

در واقع اولین و مهمترین کار در پردازش تصویر، تبدیل آن به حالتی است که قابل فهم برای کامپیوتر باشد برای همین آن را تبدیل به ماتریس می کنیم تا بتوانیم به هر یک از پیکسل های تصویر که هم

اکنون آرایه های ماتریس هستند دسترسی داشته و بتوانیم رو تصویر علمیاتی که می خواهیم از طریق محاسبات ریاضی و توابع از پیش تعریف شده انجام دهیم.

پردازش تصویر در MATLAB:

تصویر را به ماتریس تبدیل می کنیم.

برای خواندن تصویر در محیط MATLAB با استفاده از تابع imread که ورودی آن نام تصویر format آن و مسیر ذخیره‌سازی تصویر می‌باشد، عملیات مورد نظر را انجام می دهیم.

و نمایش تصویر توسط تابع imshow صورت می‌پذیرد که ورودی این تابع ماتریس تصویر است.

به عنوان مثال:

به تصویر را در متغیری به نام I ریخته و آن را در اختیار ما قرار میدهد. مقدار هر عنصر در این ماتریس شدت روشنایی تصویر در آن نقطه را نشان می‌دهد.

Imshow (I)
Imtool (I) $\Rightarrow$ **برای نمایش تصویر**
Imview (I)

دستور imview و Imtool نیز برای نمایش تصاویر بکار می‌روند در دستور Imtool ابزارهای مختلفی برای ویرایش تصویر در اختیار ما قرار می‌گیرد. جزئیات مثل رنگ و ...

اگر در برنامه‌ای بخواهیم چند تصویر را به طور همزمان اجرا تابع imshow تنها تصویر آخر را نمایش میدهده برای نمایش همه تصاویر از دستور figure استفاده

مثال* می‌خواهیم C,B,A را بگذاریم و نمایش دهیم.

I_1=imredd('ajpg');
I_2=imread ('b.jpg');
I_3=imread ('c.jpg');
Figure , im show (I_1);

Figure , im show (I_2);

Figure , im show (I_3);

خاکستری کردن تصویر(Gray)

رایج‌ترین مدل رنگ گرافیک خاکستری، رنگ‌ها از سه ترکیب قرمز، سبز و آبی تشکیل می‌شود که توسط این سه مولفه ترکیب رنگی زیادی پدید می‌آید این همان (RGB).

مدل رنگ Gray Scale از اهمیت ویژه‌ای برخوردار است چرا که در بیشتر کاربردها نیاز به تصویر رنگی نمی‌باشد و داشتن تنها یک تصویر خاکستری کافی خواهد بود.

یک پیکسل زمانی مقدار خاکستری خواهد داشت که مولفه‌های R,G,B مقدار یکسان داشته باشند با توجه به این تعیرف در مواردی که تصویر ورودی یک تصویر RGB است برای Gray Scale کردن آن از فرمول زیر استفاده می‌شود.

S-R(x-y)=S-G(x,y)=S-B(x,y)=

[R(x,y)+G(x,y),B(x,y)]/3

برای تبدیل یک تصویر رنگی به سطح خاکستری از تابع:

Reb2gray (ماتریس مورد نظر)

مثال: برنامه‌ای بنویسید که دو تصویر با نام‌های $test_2$,$test_1$ و پسوند GPg از ورودی خوانده در دو ماتریس خاکستری را نیز در خروجی نمایش دهد.

I_1=imred ('$test_1$.gpg');

I_2=imred('$test_2$.gpg');

S=rgb 2 gray(I_2);

Figure , im show (I_1);

Figure , im show (I_2);

Figure , im show (S);

و یا اینکه در خط S ; نگذاریم. Imshow s;

نکته: برای اینکه برنامه خود را در MATLAB ذخیره کنید در خط فرمان دستور باز کردن پنجره برای Edit را نوشته و دستورات برنامه خود را در آن پنجره تایپ و با پسوند m.اسم دخیره کنید با فراخوانی نام برنامه شما در محیط MATLAB اجرا می‌شود.

تفریق دو تصویر:

تفریق دو تصویر اندازه بدین منظور است که شدت روشنایی پیکسل‌های متناظر در دو تصویر را از هم، کم کنیم. فرض کنید می‌خواهیم تغییرات مغز افرادی را بررسی کنیم که دچار بیماری آلزایمر هستند برای این منظور می‌توانیم تصویری از یک مغز سالم را در مراحل مختلف با تصویر مغز فردی که دچار بیماری آلزایمر است مقایسه کنیم با اعمال عملگرد فوق بر روی دو تصویر نقاطی از مغز که در آن نقاط مغز دچار تغییر شده است مشخص می‌شود. هنگام تفریق مقادیر پیکسل‌ها، مقادیر منفی به صفر تبدیل می‌شود به عنوان مثال از تفریق دو تصویر می‌توان به شناسایی حرکت در سیستم‌های دوربین مداربسته اشاره کرد. زمانی می‌گوئیم حرکت رخداده است که بین دو فریم متوالی از دوربین اخلاف وجود داشته باشد بنابراین با تفریق fream فعلی و منیزیم قبلی اختلاف موجود بین دو تصویر را پیدا کرد و کاربرد دیگر تفریق خدمت پشت زمینه ثابت از یک تصویر است.

برای تفریق دو تصویر هم اندازه از تابع زیر استفاده می‌شود.

Im subtract (I_1,I_2)

مثال: برنامه‌ای بنویسید دو تصویر را از ورودی خوانده سایز هرکدام از آنها را بدهد دو تصویر را از هم تفریق کرده-تصویر اول و تصویر تفریق شده را نمایش دهد.

Imsize (اسم ماتریس)

تعداد سطر و ستون را میدهد.

I_1=imred ('test$_1$.gpg');

I_2=imred('test.gpg');

Imsize (I_1);

Imsize (I_2);

I3=imsubtract (I_1,I_2);

Figure , im show (I_1);

Figure , im show (S);

جمع دو تصویر:

جمع دو تصویر به این مفهوم است که در دو تصویر شدت روشنایی های پیکسل‌های متناظر با هم جمع می‌شوند یکی از رایجترین، کاربردهای جمع دو تصویر افزودن پشت زمینه به تصویر است برای این منظور نیاز به دو تصویر داریم: یکی پشت زمینه و یکی تصویر شی، تصویر زیر دو تصویر هم اندازه را با هم جمع می‌کند. اگر هم اندازه نباشد جمع و تفریق نمی‌شود باید resize کرد بعداً می‌خوانیم.

Imadd(ماتریس تصویر اول ، ماتریس تصویر دوم);

مثال: دو تصویر a_1 و a_2 را از ورودی بخوانید و سپس a_2 را از a_1 کم کنید و سپس به تصویر جدید I پشت زمینه سیاه را اضافه کرده و تصویر را نمایش دهید.

I_1=imred ('a.gpg');

I_2=imred('b,gpg');

I3=imsubtract (a,b);

[x,y]=Imsize (I_1);

Imsize (I_3);

I_3=zero(x.y);

I5=imadd(I_4,I_3)

im show (I_5);

سایز I3 را باید مشخص کنیم تا بتوانیم یک رنگ سیاه با همان سایز Q3 داشته باشیم که یک بردار [x,y] در نظر می‌گیریم.

با zeros ماتریس می‌سازیم که برابر ماتریس I_3 باشد.

مکمل کردن تصویر:

در یک تصویر سیاه و سفید که فقط دو سطح شدت روشنایی صفر و یک دارد منظور از مکمل کردن تصویر یک کردن پیکسل‌ها با مقدار صفر، و صفر کردن پیکسل‌های با مقدار یک است. در تصویر که از L سطح روشنایی برای نمایش شدت روشنایی پیکسل‌ها استفاده می‌کند مکمل یک پیکسل از فرمول زیر بدست می‌آید.

Pixel(I,j)=L-1-pixel (I,j)

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

2-1-0=1

2-1-1=0

درباره بین 255>x>0، ۲۵۶ سطح داریم در نتیجه:

$$\begin{bmatrix} 128 & 18 \\ 1 & 20 \end{bmatrix}$$

258-1-128

فرض کنید می‌خواهیم یک ربات تغییر خط با استفاده از دوربین ایجاد کنیم همانطور که می‌دانید یکی از قوانین ربات تعقیب خط مکمل شدن رنگ خط و پیش زمینه در مقاطعی از مسیر است برای مرتفع کردن این شرط می‌توان از عملگر مکمل تصویر استفاده کرد.

دستوری که ماتریس را مکمل می‌کند.

Imcomplement (ماتریس)

یک تصویر از ورودی با نام test1.tif دریافت کرده آن را مکمل کرده و تصویر ابتدایی و مکمل شده را در خروجی نمایش دهد.

I_1=imread ('test1.tif');

I_2=imcomplement(I3)

Figure imshow (I_1)

Figure imshow (I_2)

میانگین‌گیری تصویر:

فرض کنید چند تصویر یکسان داریم که بر روی هر ک دام از آنها نویزهای مختلفی وجود دارد و می-خواهیم کیفیت این تصاویر را ارتفاع دهیم در چنین مورادی می توان از همه تصاویرف میانگین بگیریم بدین ترتیب که پیکسل‌های متناظر در همه تصاویر را با هم جمع و سپس به تعداد کل تصاویر تقسیم کنیم بدیهی است که هر قدر تعداد تصاویر بیشتر باشد تصوی رحاصل از میانگین به واقعیت نزدیکتر است.

سوال: دو تصویر یکسان که دارای نویز هستند را از ورودی دریافت کرده و کیفت آن را بهبود ببخشید. (استفاده از میانگین‌گیری).

I_1=imread ('test2.jpg');

I_2=imread ('test2');

I_3=imadd (I_1,I_2)/2 ⟵ به تعداد تصاویر میانگین

Figure,imshow (I_3)

مثال برنامه‌ای بنویسید که تصویر test2 را در ماتریسی به نام I بریزد و رنگ این دو پیکسل را نمایش دهد. (۱۰۵، ۱۰۰)- (۵، ۲۰۰).

I=imread ('test2.jpg');

I(100 , 105)

(200 , 5)

مثال: تصویر زیر را در خروجی ایجاد کنید (۲۵۶، ۲۵۶)

I_1=Rand (256 , 256)

I_2=Rond (I_1)

Figure, imshow (I_1)

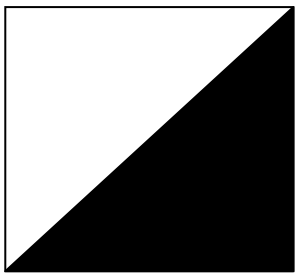
Figure, imshow (I_2)

دو تا متغیر row سطر و colums ستون

از رندوم استفاده می کنیم اما (رنگ خاکستری) پس کاری که صفر و یک شود.

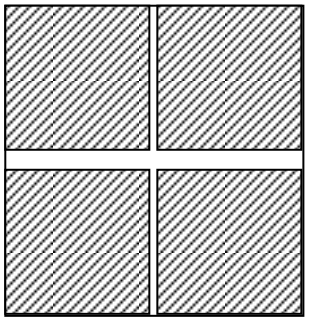
از rond هر چیزی نیم صفر بالای نیم می شود.

تصویر زیر را در خروجی ایجاد کنید.



می توانیم به این صورت هم به صورت روبرو بنویسیم مثلاً با zero کل صفحه را سیاه بگیریم به تعداد iها کم می شود هرمرحله

```
Row =256;
Clou=256;
Mat(i=row,i:col)=0
Zeros= (row,col)
For (i=1  :row
For (J=1 : col-I;)
Mat (I,j)=1
End;
End;
```



```
Row=256;
Col=256
Mat(1:row, 1:col)=0;
Zeros=(row,col)
Mat (100,105,:)=0.5
Mat (:,100:105)=1;
Imshow (mat);
```

با حلقه for

```
For i=100:105;
For j:0:col
M(I,j)=1;
End;
End;
```

For i=1 : row;

For j=100:105;

M(I,j)=0/5;

End;

End;

تغییر نوع تصویر:

در بسیاری از موارد ممکن است لازم باشد نوع تصویر تغییر یابد به عنوان مثال هنام فیلتر کردن یک تصویر index شده لازم است ابتدا آن تصویر را به یک تصویر رنگی تبدیل کرده (RGB) و سپس مراحل پردازش بعدی را روی آن انجام دهیم. دستورات که کاربرد بیشتر دارند به قرار زیر هستند:

1)gray 2 ind تبدیل تصویر خاکستری به index

[x,map]=gray 2indx(I)

ورودی‌اش یه ماتریس I هست که آن را به ماتریسی که indexها در آن هستند تبدیل می‌کند.

مثلاً INDEX $\rightarrow$ 18

را فقط به 18 index تبدیل می‌کند

تصویر اولیه index هست یعنی ماتریس دارد و هم map یعنی به تصویر index شده را به یک ماتریس سطح خاکستری تبدیل می کند.

2) ind 2 gray

Y= im2 gray(x,map)

3)y=im2bw(x,T)

هر تصویر با هر نوعی را به یک تصویر سیاه و سفید یا دودویی (bw) تبدیل می کند.

4)x=ind 2 rgb (x,map)

یک تصویر index شده $\leftarrow$ تبدیل به تصویر RGB

5)[y,map]=rgb 2 ind(x)

یک تصویر رنگی index ← RGB می‌شود.

6)y=rgb 2 gray (x)

یک تصویر رنگی RGB ← به سطح خاکستری:

7)y=mat 2 gray

هر ماتریسی که دریافت کند تبدیل به ماتریس سطح خاکستری میکند.

دستور ۳ با استفاده از آستانه T، تصویر x را به یک تصویر دودویی تبدیل می‌کند، تصویر دودویی y به ازای آن دسته از پیکسل‌های x که مقدارشان کمتر از آستانه t است دارای مقدار صفر و برای بقیه پیسکل‌ها دارای مقدار ۱ می‌شوند.

مقدار t باید فارغ از کلاس داده ورودی بین صفر و یک تعیین شوند در صورتی که دستور فوق به صورت

y=im2bw(x) * از دستور ۳ به صورت

جعبه پردازش دقیق ابزار Image processing tool book (IPT)

از تعداد ۰/۵ برای پیش فرض استفاده می‌کنیم.

مثال: ماتریس زیر به صورت زیر است برنامه‌ای بنویسید که این ماتریس را با سطح آستانه ۰/۶ تبدیل به یک تصویر دودویی کند و تصویر را در خروجی نمایش دهد.

```
F=[1   2;   3    4]
Figure, imshow(f);
Y=im2bw(f,0/6)
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$1 > 0.6 \Rightarrow 1$

$2 > 0.6 \Rightarrow 2$

.

.

.

$$\Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

G=mat 2 gray (f) ماتریس به سطح خاکستری تبدیل چون باید بین یک و صفر باشد.

```
Figure,imshow (g);
Y=im2bw(g)
Figure,imshow(y);
```

حال تصویر خاکستری را بهشان بده مثلاً مقادیر زیر:

$$\begin{bmatrix} 0.5 & 0.33 \\ 0.2 & 0.6 \end{bmatrix} \Rightarrow \boldsymbol{T} = 0.6$$

$$0.5 < 0.6 \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$0.6 = 0.6 \Rightarrow 1$$

هیستوگرام

هیستوگرام یا نمودار فراوانی سطوح خاکستری، معیاری از میزان تکرار و پخش مقادیر سطوح روشنایی در محدوده‌ی صفر تا ۲۵۵ روی کل تصویر است.

هیستوگرام یک تصویر یکی از مهمترین خصوصیات آن در حوزه‌ی پردازش تصویر به حساب می‌آید. و عملیاتی مانند بهبود contrast (و جنوح)، استخراج ویژگی و یافتن آستانه‌های بهنیه برای باینری کردن تصویر مورد استفاده قرار می گیرد. برای تشخیص هسیتوگرام، تعداد پیکسل‌های تصویر که مقدارشان صفر تا ۲۵۵ است شمرده می شود و برحسب سطوح خاکستری رسم می شود در جعبه ابزار پردازش تصویر می‌توان به کمک دستوراتی مانند imhist، هیستوگرام تصویر را مشاهده نمود و سپس با اجرای فرامین histeq و imaddjust هیستوگرام تصویر را یکنواخت یا به نحو دلخواه تنظیم کرد.

مثال: برنامه‌ای بنویسید که یک تصویر با نام p.tif را از ورودی دریافت کرده هسیتوگرام تصویر را نمایش داده و سپس به منظور افزایش کنتراست تصویر هیستوگرام یکنواخت شده و تصویر حاصل از این هیستوگرام رادر خروجی نمایش دهد.

```
I=imread ('p.tif');
```

```
Imhist(I);
J=histeq(I);
Imhist (J);
Imshow(J);
```

هیستوگرام که یک طرف جمع شده پخش می کند (باعث وضوح تصویر می شود)

تبدیل تصویر سطح خاکستری به index:

برنامه ای بنویسید که یک تصویر سطح خاکستری با نام camera.tif از ورودی دریافت و آن را به یک تصویر index شده با index.18 تبدیل کرده و هر دو تصویر را در خروجی نمایش دهید.

```
I=imread('camera.tif')
Figure(1), imshow I
[x,map]=gray 2 ind (I,18)
Figure (2), imshow (x,map)
```

تبدیل تصویر به باینری و استفاده از آستانه.

برنامه‌ای بنویسید که یک تصویر را از ورودی دریافت کرده و آن را به تصویر سیاه و سفید یا بابنری تبدیل و هر دو تصویر را در خروجی چاپ cobmp.

```
I=imread ('c.bmp')
Figure (1),imshow(I)
G=mat2 gray(I)
*figure,imshow (g);
Y=im2bw (g,0.5);
Figure (2), imshow (y)
```

*اگر می خواست تصویر خاکستری هم نمایش این را می نوشتیم.

برنامه‌ای بنویسید که یک تصویر رنگی index شده با نامن f.tif را از ورودی دریافت و آن را تبدیل به سطح خاکستری نماید و در نهایت هر دو تصویر را نمایش دهد.

```
I=imread ('f.tif.jpg')
J=ind2gray (I.map)
Figure (1), imshow (I)
Figure (2),imshow (J)
```

برنامه‌ای که یک تصویر RGB را از ورودی خوانده و به یک تصویر index شده با ۱۵ اندیش تبدیل کند و در نهایت هر دو تصویر را نمایش دهد.

```
I=imread ('f.tif');
[x,map]=rgb 2 ind (I,15);
Figure (1), imshow (I);
Figure (2),imshow (x,map);
```

تبدیلات حوزه مکان: Spacial Transform

شامل تغییر اندازه تصویر، بزرگ نمایی و کوچک نمایی، برش بخشی از تصویر،انتقال و دوران می باشد همانگونه که میدانید برای کوچک نمایی یک تصویر باید تعدادی از نمونه های آن دور ریخته شود و هنگام بزرگنمایی و تعداد پیکسل‌های تصویر افزوده شود. مقادیر این پیکسل‌ها معمولاً با روشهای درون یابی و با استفاده از قانون همسایگی تعیین می شود.

تغییر ابعداد یک تصویر:

دستور imresize برای تغییر اندازه تصویر به کار می رود آرگومانهای ورودی این دستور ماتریس تصویر اولیه و میزان بزرگ نمایی یا کوچک نمایی آن است. علاوه بر تعیین نسبت تغییر ابعاد در این دستور می توان مستقیماً ابعداد را وارد کرد.

مثال:

برنامه ای بنویسید که یک تصویر را از ورودی خوانده آن را ۱,۲۵ برابر کرده در خروجی نمایش دهد و بار دیگر آن تصویر را اندازه ۳۰۰*۲۰۰ نموده و آن را در خروجی نمایش دهد.

```
I=imread ('p.tif');
I1=imresize(I,1.25);
I2=imresize(I1,[200,300]);
Figure (1), imshow (I);
Figure (2),imshow (x,map);
```

دوران یک تصویر: دستور imrotate برای دوران یک تصویر به اندازه دلخواه بکار می رود هنگام استفاده از این دستور مقدار زاویه چرخش برحسب درجه تعیین می شود زاویه مثبت منجر به چرخش در جهت خلاف عقربه‌های ساعت و زاویه منفی منجر به چرخش در جهت عقربه‌های ساعت می‌گردد.

مثال: برنامه ای بنویسید که یک تصویر را دریافت کرده و آن را با درجه ۳۵ در جهت عقربه‌های ساعت دوران داده و تصویر را نمایش دهد.

آرگومان‌های ورودی ماتریس تصویر و زاویه دوران است.

```
I1=imread('a.jpg');
I2=imrotate(I,-35);
Figure, imshow (I);
Figure,imshow (I1);
```

برش تصویر: دستور imcrop برای برش یک بخش مربعی از تصویر مورد استفاده قرار می گیرد برای این کار می توان به صورت مستقیم با کمک mos بخش مورد نظر را انتخاب کرد، یا اینکه با اعلام مختصات نقاط موردنظر به ناحیه برش یافته دست پیدا کرد.

دستور imcrop تصویر را بعنوان پارامتر ورودی می‌گیرد و پس از اجرای آن پنجره‌ای باز می‌شود که تصویر در آن نمایش داده شده و منتظر است تا ناحیه موردنظر را تعیین کنید مکان نما بر روی تصویر به شکل + است و منتظر click شماست.

آرگومان ورودی این دستور، ماتریس تصویر است، اما اگر بخواهیم با مختصات این کار را انجام دهیم ترتیب پارامترهای بخش برش یافته به صورت زیر است.

مثال: برنامه ای که یک تصویر را از ورودی گرفته امکان برش از طریق موس را به ما داده و در مرحله بعد با شروع از نقطه (۴۰، ۶۰) و با طول ۱۰۰ و عرض ۹۰ تصویری ایجاد کرده و آن را در خروجی نمایش دهد.

این دستور یا شامل تنها آرگومان ماتریس تصویر است و یا علاوه بر ماتریس تصویر آرگومان مختصات نقطه را نیز دارا می‌باشد.

```
I=imread('a.jpg');
I1=imcrop(I);
```

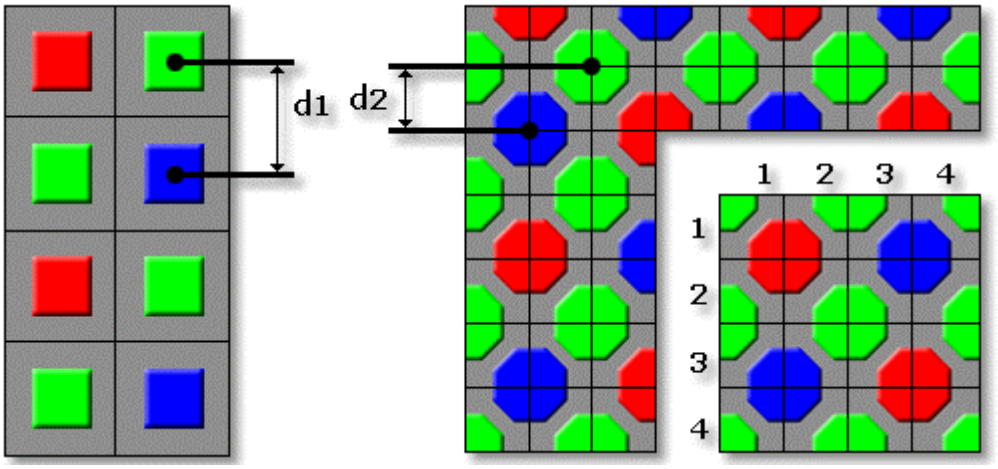
Figure, imshow (I);

I_2=imcrop(I1,[40 60 90 10]);

Figure,imshow (I_2);

برخی روابط اساسی بین پیکسل‌ها:

همسایه‌های یک pixel، پیکسلی به نام p با مختصات p(x,y)، ۴ همسایه افقی و عمودی دارد.



این پیکسل‌ها همسایه ۴ گانه p نامیده می شوند که آن را با $N_4(P)$ نمایش می دهند.

هر پیکسل یک فاصله واحد از (X,Y) دارد اگر پیکسل X,Y بر روی مرز تصویر دیجیتال قرار بگیرد بعضی از همسایه‌های آن خارج تصویر قرار می‌گیرند. (مثل درست کردن در تصویرهای قدیمی)

۴ همسایه قطری P مختصاری به شرح زیر دارند:

۴ همسایه قطری P را به صورت $N_D(P)$ نمایش می دهند.

۴ همسایه قطری و ۴ همسایه افقی عمودی در مجموعه- همسایه ۸ گانه را تشکیل می دهند که آن را با $N_8(P)$ نمایش می دهند.

$16[N_4(P)+N_D(P)=N_8(P)]$

اتصال، هنگامی دو pixel به هم متصل هستند که همسایه باشند و سطح خاکستری آنها با هم شرط مشخصی را برآورده کند اتصال اساس تشکیل مرز و ناحیه است. ارتباط بین پیکسل‌ها یک مفهوم

اساسی است که مفاهمی مانند مرز و مناطق را ساده می‌کند. اگر دو پیکسل در ارتباط باشند باید ببنیم آیا همسایه‌اند یا اینکه مقدار سطح خاکستری‌شان یکسان است.

مثلاً در تصاویر باینری با ارزش صفر ، یک دو پیکسل ۴ همسایه دارند اما زمانی می توان گفت این دو pixel با هم مرتبط هستند که ارزش آنها یکسان باشد.

بهبود تصویر: وقتی راجع به مقادیر سطح خاکستری صحبت می کنیم بازه ما بین ۲۵۵-۰ است در بسیاری از عملیات پردازش تصویر این بازه، این بازه، به بازه صفر ، یک نگاشت می شود.

بهبود تصویر چیست؟

بهبود تصویر فرآیند ایجاد یک تصویر قابل استفاده برای کاربر است دلایل استفاده از بهبود تصویر عبارت است از:

۱-هایلات جزئیات موردنظر در تصاویر

۲-حذف nois از تصاویر

۳-ایجاد تصویر با ظاهر بصری بالا

نکته: مات کردن تصویر گاهی در جهت، بهبود تصویر به کار می رود مثلاً زمانی که می خواهیم چین و چروک را از تصویر چهره حذف کنیم و یا در صنعت چاپ برای حذف فضای خالی بین حروف

بهبود تصویر و پردازش هسیتوگرام تصویر:

هیستوگرام یک تصویر توزیع ستوح، خاکتسری، در تصویر را نمایش می دهد. و در پردازش تصویر خصوصا مدار منطقی مفیدمی باشد.

توجه کنید که تصاویر با کنتراست بالا دیاگرامی با فاصله های تقریبا مساوی دارد.

انواع دیاگرام

فیلترها و بهینه سازهای تصویر:

نمودار هیستوگرام Histogram : این نمودار تنها برای تصاویر خاکستری و جهت نمایش میزان کنتراست تصویرنسبت به شدت نور تابیده شده به تصویر به کار می رود.این نمودار بهترین روش برای مقایسه ۲ تصویر از یک نما خواهد بود دستور آن (ماتریس خاکستری)imhist می باشد.توجه کنید که این دستور مانند دستورimshow نیازی به متغیر ندارد و در خروجی هیستوگرام خروجی را نمایش میدهد.

مثال:

برنامه ای بنویسید که یک تصویر رنگی را از ورودی دریافت کرده و سپس میزان روشنایی تصویر را به اندازه ۸۰ واحد افزایش داده در خروجی تصویر اولیه ،تصویر بهبود یافته از لحاظ روشنایی و هیستوگرام را در یک پنجره نمایش دهد.

```
A=imread('m1.jpg');
B=imadd(a,80);
C=rgb2gray(a);
Subplot(2,2,1),imshow(a)
Subplot(2,2,2),imshow(b)
Subplot(2,2,3),imhist(c)
Subplot(2,2,4),imshow(c)
```

فیلتر حد فاصل:

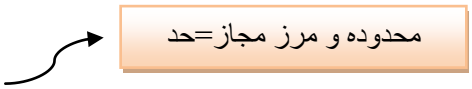
Imcontour(ماتریس خاکستری,n)

دستور بالا زمانی به کار میرود که شما نیاز دارید پستی و بلندی تصویر را با خطوطی مشخص کنید ، مصلا در تصویر های هوایی این فیلتر کاربرد فراوانی دارد.ورودی این دستور یک ماتریس و تعداد رنگ های مرز می باشد .اگر تعداد رنگ را مشخص نکنیم پیش فرض ۳ در نظر گرفته می شود.(مثال بالا را در این زمینه کامل کنید)

```
A=imread('m1.jpg');
B=imadd(a,80);
C=rgb2gray(a);
Subplot(2,2,1),imshow(a)
Subplot(2,2,2),imshow(b)
Subplot(2,2,3),imhist(c)
Subplot(2,2,4),imshow(c)
Subplot(2,2,4),imcontour(c,4)
```

تنظیم کننده ی خودکار کنتراست تصویر:

Imadjust (تصویر خاکستری)



Imadjust (ماتریس رنگی , stretchlim (ماتریس رنگی) , [ ])

با استفاده از تابع imadjust کنتراست یا وضوح تصویر به طور خودکار تنظیم می شود.اگر تصویر خاکستری باشد کافی است تابع imadjust را بر روی آن اعمال کرده در یک متغیر جدید بریزیم و سپس آن را نمایش دهیم اما اگر تصویر رنگی باشد تابع stretchlim حد مجاز برای کنتراست تصویر را تعیین کرده و اطلاعات را در اختیار تابع imadjust قرار میدهد.

مثال:

برنامه ای بنویسید که ۲ تصویر را ، (یکی رنگی و دیگری خاکستری) از ورودی دریافت کرده آنها را به طور خودکار از لحاظ وضوح تنظیم کرده و هر ۴ تصویر را در خروجی نمایش دهد.

```
A=imread('m1.jpg');
C=rgb2gray(a);
A1=imadjust(a,stretchlim(a),[]);
C1=imadjust(c);
Subplot(2,2,1),imshow(a)
Subplot(2,2,2),imshow(c)
Subplot(2,2,3),imshow(a1)
Subplot(2,2,4),imshow(c1)
```

دستور histeq :

در تصاویر خاکستری با استفاده ازهیستوگرام تصویر می توانیم کیفیت شدت نور را بهینه کنیم .به این صورت که دستور histeq را بر روی ماتریس تصویر خاکستری اعمال کرده و در متغیر جدیدی ذخیره میکنیم .حال اگر هیستوگرام تصویر را به نمایش در آوریم خواهیم دید که histogram بهبود یافته است .

اضافه کردن نویز(noise) به تصویر :

همیشه بهینه کردن و بهبود بخشیدن یک تصویر هدف کاربر نیست.گاهی نیاز است که یک تصویر با نویز و اختشاش وارد سیستم شود.۳ نوع نویز که کاربرد بیشتری دارند به شرح زیر است :

۱ -**نویز Gaussian** (نویز سفید با میانگینی از کنتراست تصویر)

۲ -**نویز salt & pepper** (نویز فلفل – نمکی)(اگر تصویر خاکستری باشد فلفل نمکی سیاه سفید است)(اگر تصویر رنگی باشد فلفل نمکی سبز و قرمز خواهد بود)

۳ -**نویزSpeckle** (نویز لکه ای یا نویزی که باعث افزایش میزان اختلال در تصویر می شود)

(میزان نویز,'نوع نویز',ماتریس)imnoise=متغیر جدید

نکته : میزان نویز عددی بین صفر تا یک است.

مثال:

برنامه ای بنویسید تصویر رنگی دریافت کرده و هر سه نوع نویز را با میزان دلخواه خودتان بر تصویر اعمال نموده و در یک پنجره تصویر اولیه و ۳ تصویر با نویز را نمایش دهد؟

```
A=imread('m1.jpg');
A1=imnoise(a,'gaussian',0.2);
A2=imnoise(a,'salt & pepper',0.4);
A3=imnoise(a,'speckle',0.8);
Subplot(2,2,1),imshow(a)
Subplot(2,2,2),imshow(a1)
Subplot(2,2,3),imshow(a2)
Subplot(2,2,4),imshow(a3)
```

فیلتر حذف نویز از تصویر:

این فیلتر باعث می شود که انواع نویز های موجود در یک تصویر تا حد قابل قبولی حذف شود و تصویری شفاف تر در اختیار داشته باشیم

روش از بین بردن نویز: در این دستور به این صورت است که سیستم چند پیکسل کنار یکدیگر را بررسی کرده و رنگ میانگین را جایگزین پیکسل های دارای نویز میکند.

متغیر= medfilt2(ماتریس)

نکته : این دستور بر روی تصاویر خاکستری عمل میکند.

مثال :

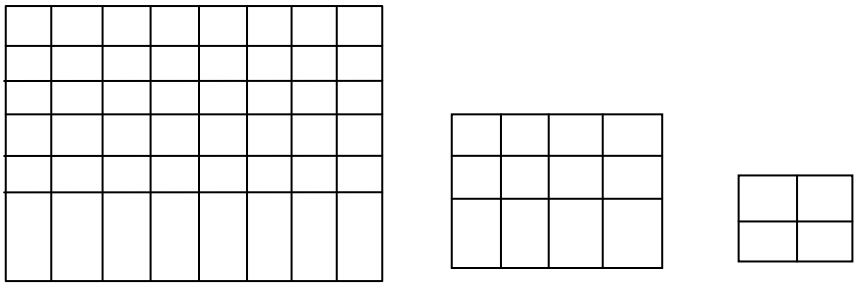
برنامه ای بنویسید که تصویری از ورودی دریافت کرده و نویز لکه ای با میزان ۰,۲ به آن اعمال نموده و سپس از تصویر دارای نویز حذف نویز کرده و هر سه تصویر را در خروجی نمایش دهد؟

```
A=imread('m1.jpg');
B=rgb2gray(a);
C=imnoise(b,'speckle',0.2);
D=medfilt2(c);
Subplot(1,3,1),imshow(a)
Subplot(1,3,2),imshow(c)
Subplot(1,3,3),imshow(d)
```

فیلتر حذف نویز تطابقی :

روش انجام فیلترینگ در این تابع به این صورت است که کاربر تعداد پیکسل های شبکه بندی شده ی یک تصویر را در نظر میگیرد (مقادیر x,y). با این روش تصویر به صورت یک شبکه تنظیم میشود که هر شبکه ماتریسی است که شما تعداد سطر و ستون آن را تعیین نموده اید.تابع هر ماتریس را مورد بررسی قرار میدهد و به صورت تطابق رنگ یک رنگ را برای پیکسلی که دارای رنگی متفاوت با سایر پیکسل های اطراف خود است مشخص می کند و آن را به رنگ پیکسل های اطراف خود در می آورد.

Wiener2 (ماتریس , [x , y])



نکته: فیلتر ها معمولا به شکل مربع می باشند.

مثال :

برنامه ای بنویسید که یک تصویر را از ورودی دریافت کرده و سپس نویز فلفل نمکی با میزان ۰,۴ بر آن اعمال نموده و سپس توسط فیلتر تطالبقی با اندازه ۴*۴ از آن حذف نویز نموده، هر سه تصویر را در خروجی نمایش داده ؟

```
A=imread('m2.jpg');
B=rgb2gray(a);
C=imnoise(b,'salt & pepper',0.4);
D=wiener2(c,[4,4]);
Subplot(1,3,1),imshow(a)
Subplot(1,3,2),imshow(c)
Subplot(1,3,3),imshow(d)
```

ایجاد فیلتر دلخواه و طراحی آن:

شاید هیچکدام از این فیلتر ها شما را به هدفتان نرساند و بخواهید خودتان فیلتری را طراحی کنید .

تابع fspecial(type) : به شما کمک میکند تا یک فیلتر را طراحی کنید .برای استفاده از این فیلتر بر روی یک تصویر باید فرم و الگوریتم استفاده آن را مشخص کنید.مثلا ممکن است الگوریتم آن ماتریس مربعی و یا دایره ای باشد.

فرم های استفاده از فیلتر:

type	توصیف
'average'	فیلتر با روش میانگین، ماتریس مربعی
'disk'	فیلتر با روش میانگین، ماتریس دایره ای
''gaussian'	فیلتر پایین گذر gaussian
'laplacian'	فیلتر لاپلاس
'log'	فیلتر لاپلاس با حذفGaussian
'motion'	فیلتر حرکت دهنده
'prewitt'	فیلتر تقویت لبه
'soloel'	فیلتر لبه افقی و عمودی
'unsharp'	فیلتر افزایش شدت نور

طراحی فیلتر میانگین با ماتریس مربعی:

Fspecial('average',x)

برای فیلتر کردن یک تصویر نمی توانید تمامی تصویر را یک جا فیلتر نمائید باید تصویر را به صورت شبکه های کوچکی تقسیم نموده و هر شبکه را تک تک فیلتر کرده تا عملیات فیلترینگ بر روی تمام تصویر انجام شود .به همین دلیل برای اینکه از یک فیلتر استفاده کنید کافی است یک شبکه ماتریس مربعی در اندازه دلخواه طراحی کرده و بعد با تابعی خاص از این فیلتر استفاده نمائید. اولین فیلتر ، فیلتر میانگین است که باعث محو شدن تصویر می شود هر چه مقدار x در الگوریتم average بیشتر باشد شبکه ماتریس فیلتر بزرگتر و در نتیجه تصویر محو تر می شود.

استفاده از یک فیلتر طراحی شده :

متغیر=imfilter (نام فیلتر,ماتریس)

این تابع برای استفاده از فیلتری که شما طراحی کرده اید بر روی تصویر مورد نظر به کار می رود. بعد از طراحی فیلتر آن را در متغیری ذخیره کنید و سپس توسط این دستور آن را بر تصویر اعمال نمائید .

مثال:

برنامه ای بنویسید که یک تصویر از ورودی دریافت کند و سپس یک فیلتر محو کننده با اندازه 4x4 تولید نموده ، نام این فیلتر را Blur گذاشته و سپس فیلتر را بر تصویر مورد نظر اعمال نموده ، هر دو تصویر را در خروجی نمایش دهد.

```
A=imread('m4.jpg');
Blur=fspecial('average',4);
D=imfilter(a,blur);
Subplot(1,2,1),imshow(a)
Subplot(1,2,2),imshow(b)
```

طراحی فیلتر میانگین با ماتریس گرد:

الگوریتم این فیلترها مانند الگوریتم فیلتر average است

ماتریس این فیلتر به صورت دایره عمل میکند

X مقدار شعاع دایره است

متغیر= fspecial (‘disk’)

طراحی فیلتر پایین گذر گوسین:

سیگنال ویدئو در سیستم های ویدئویی فرکانسی بین صفر تا پنج مگاهرتز دارد.

فرکانس های پایین باعث ایجاد اسکلت کلی و قالب تصویر می شوند.

فرکانس های بالا باعث ایجاد لبه ها و جزئیات تصویر می شوند.

متغیر=fspecial(‘gaussian’,x,s)

اندازه ماتریس شبکه فیلتر=X

مقداری با نام سیگما-که میزان فرکانس را تعیین می کند=S

طراحی فیلتر لاپلاس:

متغیر= fspecial(‘laplacian’,a)

این فیلتر یک قالب از تصویر مد نظر شما را نمایش می دهد.اندازه ی این فیلتر 3x3 طراحی شده و غیر قابل تغییر است .

طراحی فیلتر لاپلاس از روش حذف گوسین:

مانند الگوریتم laplacian عمل می کند.اما اندازه و شبکه فیلتر قابل تیغییر است و ماتریس دایره ای است .

متغیر= fspecial(‘log’,x,s)

طراحی فیلتر حرکت کننده:

لرزش دست شما هنگام عکاسی ، خالی را بر روی تصوزر ایجاد میکند . در واقع این کار را فیلتر حرکت دهنده نیز انجام می دهد .

متغیر= fspecial(‘motion’,len,theta)

تعیین شدت حرکت=Len

تعیین زاویه حرکت=Theta

۹۰ درجه= حرکت بالا و پائین
۰ درجه =حرکت چپ و راست

طراحی فیلتر تقویت لبه:

لبه های تصویر را نمایان می کند .

چهت نور پردازی اجسام تصویر را مشخص می کند

متغیر= fspecial(‘prewitt’)

طراحی فیلتر مشخص کننده لبه افقی و عمودی :

لبه های افقی و عمودی را مشخص می کند

متغیر= fspecial(‘sobel’)

طراحی فیلتر افزایش دهنده شدت نور در لبه ها

این فیلتر بر خلاف average عکمل میکند.

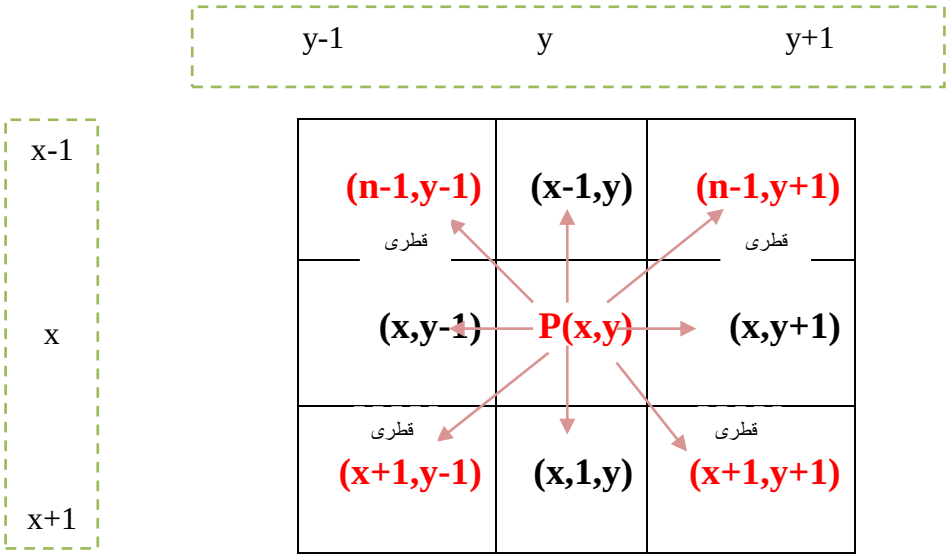
باعث افزایش شدت نور در تصویر و لبه های آن می شود .

متغیر= fspecial(‘ansharp’,a)

A= میزان شدت نور عددی بین صفر و یک

شناسایی اشیاء :

همسایگی چهار گانه :

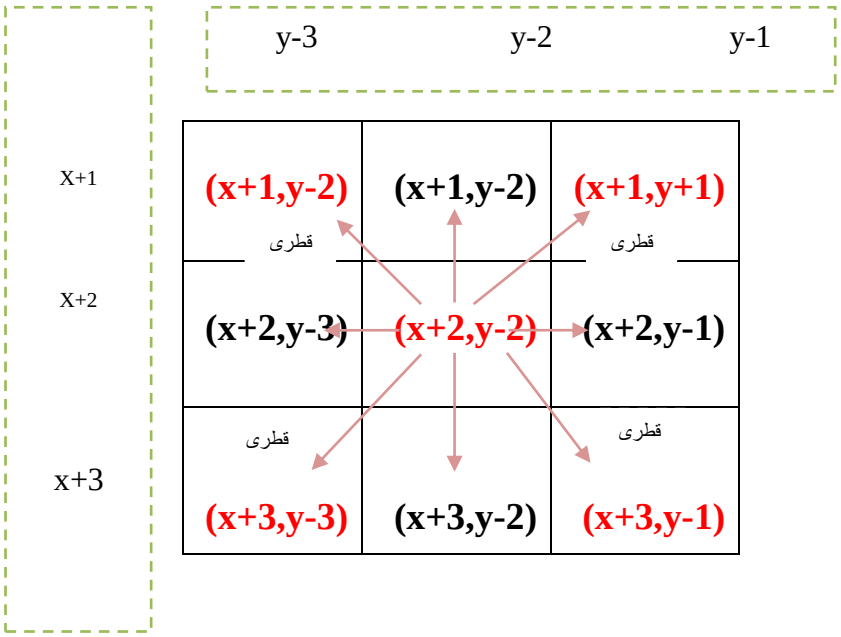


همسایگی یک پیکسل :

پیکسلی با نام p و با مختصات x,y چهار همسایه افقی و عمودی دارد. این پیکسل ها را همسایه چهارگانه می نامند.همچنین نقطه p دارای پیکسل های قطری نیز می باشد. مجموع چهار همسایگی افقی عمودی و چهار همسایگی قطری را همسایگی هشت گانه می نامند.

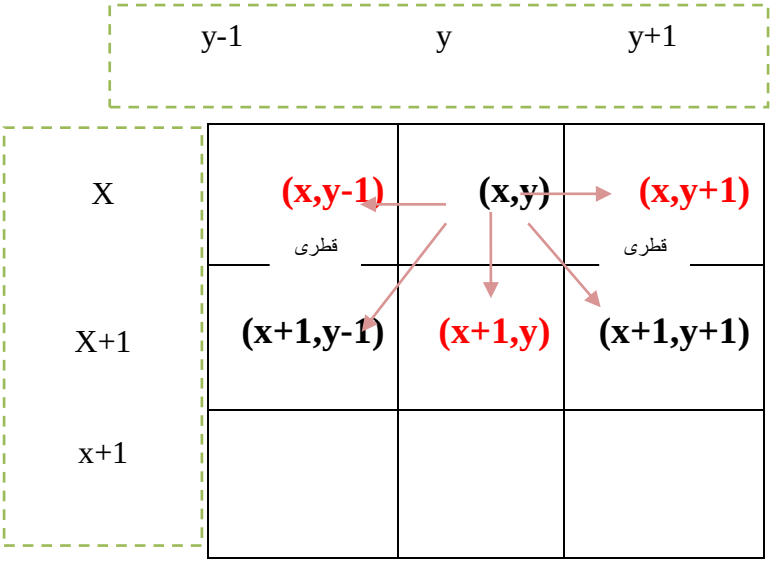
مثال :

همسایگی هشت گانه نقطه a با مختصات (x+2,y-2) را رسم کنید.



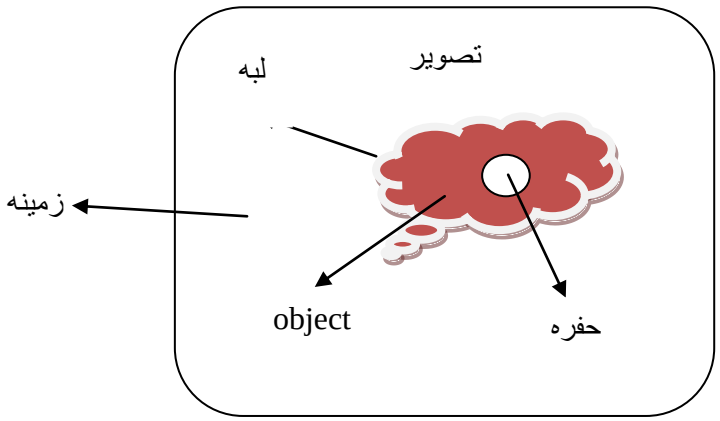
مثال:

پیکسلی با مختصات x,y دارای سه همسایه افقی و عمودی و دو همسایبه قطری می باشد.کل همسایگی آن را رسم کنید .



تعریف لبه :

به نقاطی از تصویر گفته می شود که در آن نقاط ۲ پیکسل کنار هم ۲ مقدار متفاوت داشته و یا آن دو مقدار از نظر ارزش عددی فاصله زیادی با هم داشته باشد .لبه ها در حقیقت مرز بین زمینه و object را در یک تصویر مشخص می کند.



چگونگی تولید هیستوگرام:

در واقع هیستوگرام نمودار فراوانی (آمار) یعنی تعداد دفعات مقدار مختلف در کل تصویر را نمایش می دهده از هسیتوگرام تصویر می توان در موضع ویژگی‌هایی همچون روشنایی تاریکی و کنتراست تصویر صحبت کرد.

هیستوگرام تعدیل:

اگر تصویر ما دارای هیستوگرامی باریک پائین نمودار و یا بالای نمودار باشد برای افزایش کنتراست تصویر از هیستوگرام تعدیل استفاده می شود. بدین صورت که شدت ورودی از یک تابع عبور داده شده و شدت خروجی دارای هیستوگرام تعدیل می شود این دستور عبارت است:

histeq این دستور توسط

شدت خروجی $S=T(r)$

هیستوگرام تعدیل یا کلاً هم تکنیک‌های بهبود با توجه به نیاز کاربر وسایر عوامل دخیل می تواند بهبود دهنده باشد یا خیر، گاهی برای رسیدن به یک هدف مطلوب چند تکنیک با یکدیگر ترکیب می‌شود.

بهبودتصویر به دو دسته تکنیک تقسیم می‌شود:

۱-تکنیک بهبود حوزه مکان

۲-تکنیک بهبود حوزه فرکانس

در تکنیک بهبود حوزه مکان مستقیماً با پیکسل‌های تصویر سروکار داریم اما در تکنیک بهبود حوزه فرکانس با تعدیل فوریه یا تبدیل موجک سروکار داریم.

تبدیل فوریه یک سری ضرایب از تصویر را به ما میدهد که این ضرایب این ویژگیهای تصویر محسوب می‌شوند پردازش نقطه‌ای در حوزه مکان point prossosing

تکنیک‌های پردازش نقطه‌ای عبارتند از: منفی سازی تصویر یا negative، آستانه‌گیری، تبدیل لگاریتمی، تبدیل قانون توان، بخش بندی سطح خاکستری و بخش بندی صفحه بیتی.

مبانی تبدیلات شدت و فیلتر کردن مکانی:

در این بخش تمام تکنیکهای پردازش تصویر در حوزه مکانی پیاده سازی می شود پردازش حوزه مکانی به صورت زیر بیان می شود که در آن $g(x,y)=T[f(x,y)]$

$g(x,y)$ تصویر خروجی و $f(x,y)$ تصویر ورودی و t عملگر روی f است که بر روی همسایگی x,y تعریف شده است. عملگرد یا تابع می تواند بر روی یک تصویر یا مجموعه‌ای از تصاویر اعمال شود نقطه x,y یک مکان اختیاری در تصویر است و منطقه کوچکی که حاوی x,y است همسایگی نامیده می شود این روش فیلتر کردن مکانی نامیده می شود. کوچکترین همسایگی ممکن است به اندازه یک در یک ۱*۱ باشد و در این مورد g فقط به مقدار f در نقطه x,y بستگی دارد و t در معادله، به عنوان تابع تبدیل شدت به صورت $S=T(r)$ محاسبه می شود.

به طور مثال اگر t(r) مانند شکل زیر باشد اثر اجرای تبدیل به هر پیکسل از f برای تولید pixel های متناظر از G تصویر با کنتراست بالاتر از تصویر اصلی تولید خواهد شد که برای این کار شدت پایینتر از k را تیره و سطوح شدت بالاتر از k را به رنگ روشن نمایش خواهد داد در این تکنیک که گاهی گسترش کنتراست نامیده می شود. مقادیر r<k توسط تابع تبدیل به بازه باریکی از s به سمت رنگ سیاه فشرده‌سازی می‌شود. T(r) یک تصویر دو سطحی دودویی ایجاد می کند این نوع نگاشت تابع تعیین آستانه یا آستانه گیری نام دارد.

بعضی از توابع تبدیل تابع: greylevel

تبدیلات شدت نسبت به سایه روشنهای پردازش تصویر ساده است. مقادیر pixelها قبل و بعد از پردازش به ترتیب توسط s,r نمایش داده می شود، برای معرفی تبدیلات شکل زیر را در نظر بگیرید که ۳ نوع اصلی از توابع را نشان میدهد که مکرراً برای به کار می رود.

۱-تابع خطی تبدیلات نگاتیو یا منفی و یا همانی

۲-توابع لگاریتمی و معکوس آن

۳-قانون،توان، تبدیلات ریشه n ام توان nام.

تابع همانی حالت ساده ای است که در آن شدتهای خروجی معادل شدتهای ورودی‌اند و فقط برای کامل بودن در گراف گنجانده می شوند.

نگاتیو تصاویر: نگاتیو تصویری با سطوح شدت در بازه [0,L-1] با استفاده از تبدیل نگاتیو نشان داده شده در شکل بدست می‌آید که به صورت S=L-1-r نشان داده می‌شود معکوس کردن سطوح شدت تصویر در این روش مانند نگاتیو عکس عمل می کند. این نوع پردازش معمولاً برای ارتفاع بخش سفید تصویر به خاکستری در مناطق تاریک تصویر بکار برده می شود مخصوصاً زمانی که اندازه نواحی سیاه تصویر زیاد باشد.

تبدیلات لگاریتمی: شکل کلی تبدیلات لگاریتمی به صورت زیر می باشد:

$S=c*\log(r+1)$ که در آن c عدد ثابت و $r=>0$ در نظر گرفته می شود. شکل منحنی لگاریتمی نشان میدهد که این تبدیلات بازه نازکی از مقادیر با شدت پائین در تصویر ورودی را به بازه وسیع تری از سطوح خروجی تبدیل می کند. درباره مقادیر بالاتر در ورودی عکس آن صحیحاست از این تبدیل برای بسط مقادیر، پیکسل‌های تاریک در یک تصویر و فشرده سازی مقادیر سطح بالاتر استفاده خواهیم کرد.

تبدیلات قانون توان (گاما)

تبدیلات قانون توان به صورت زیر است:

$$s = c*(r-\delta) \qquad S=c*r$$

که c و δ ثابت‌های مثبت هستند، گاهی معادله به هحالت دوم نوشته می شود تا انحرافی ایجاد کند یعنی خروجی قابل اندازه‌گیری را برای وقتی که ورودی صفر است تولید کند.

طراحی و پیاده‌سازی فیلترهای دوبعدی خطر برای داده‌های تصویر.

فیلتر کردن عملیاتی برای اصلاح و یا به سازی تصویر محسوب می‌شودبه عنوام مثال: برای تأکید بر مشخصه‌ای خاص از تصویر و یا حذف مشخصه دیگر از آن می توان تصویر را فیلتر نمود. مانند حذف فرکانس‌های بالا و نگه داشتن فرکانسهای پائین، عملیات فیلتری در حوزه‌ی پردازش تصویر شامل ۱- smoothing (نرم کردن) (عموماً با فیلترهای پائین گذر)- ۲- sharpening (تیز کردن) (اغلب با فیلترهای بالاگذر) و بهبود لبه‌ها edge enhacement (بهبود لبه‌ها) می‌باشد.

فیلتر کردن تصویر یک عملیات همسایگی است که در آن مقادیر هر پیکسل در تصویر خروجی از اعمال یک الگوریتم روی نقاط همسایه پیکسل‌های متناظر در تصویر ورودی به دست می‌آید، فیلتر کردن خطی تبدیل است که در آن از ترکیب خطی مقادیر همسایه استفاده می‌شود، مفهوم فیلترینگ

در پردازش تصویر در دو حالت حوزه مکان و حوزه فرکانس استفاده می شود هم چنین عملیات مورد استفاده در filtering، کانولوشن (convolution) کوریشن (correlation).

فیلترهای مکانی: عملیات، همسایگی گفته می شود که علاوه بر خود پیکسل با پیکسل‌های همسایه آن نیز سروکار داریم، همسایه‌ها اغلب مربع حول یک pixel هستند، ساده ترین عملیات همسایگی شامل موارد زیر است.

۱-min مقدار pixel مرکزی با کمترین مقدار از همسایگی آن جاگذاری می‌کند.

اگر خود pixel مرکزی دارای کمترین مقدار بود کدش باقی می ماند.

۲-max برعکس min

۳-median (میانه): ابتدا ۹ عدد حاصل از pixel مرکزی و همسایه‌های آن را از کوچک به بزرگ sort کرده و سپس عدد پنجم را به عنوان میانه انتخاب می کند،یکی از راه حلهای حذف نویز فلفل نمکی استفاده از medium است، وقتی از median استفاده می شد چون میانه اعداد انتخاب می شود معمولاً noise حذف می‌شود اگر naise زیاد بود چندین بار از این عمل استفاده می شود.

فرایند فیلتر مکانی: در عملیات فیلتر مکانی ما یک ماسک با ابعاد فرم داریم مانند:

یک نمونه ساده از کابرد ماسک:

a	b	c
d	e	f
g	h	i

X	Y	Z
U	V	W
L	M	N

	E p	

این عمل برای هر یک از pixelهای تصویر اصلی جهت تولیدتصویر فیلتر شده تکراری می شود.

در تصویر خروجی با استفاده از max داده شده

Eprocessed=a*x+b*y+c*z+d*u+e*v+f*w+g*l+h*m+i*n

در تصویر ورودی قرار است از mask، median استفاده شود، تصویر خروجی پس از اعمال mask midian به مقداری در pixel مرکزی‌اش قرار می گیرد؟

1	2	0			
2	2	4			
1	0	1			

	؟				

W(-1,-1)	W(-1,0)	W(-1,1)
W(0,-1)	W(0,0)	W(0,1)
W(1,-1)	W(1,0)	W(1,1)

F(x-1,y-1)	F(x-1,g)	F(x-1,y+1)
F(x,y-1)	F(x,y)	F(x,y+1)
F(x+1,y-1)	F(x+1,y)	F(x+1,y+1)

فرم معادله مکانی $$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)\,f(x+s,y+t)$$

ساده‌ترین فیلترهای مکانی هموارسازی

smoothing spacial filters

ساده‌ترین عملیات فیلتر مکانی عمل هموارسازی است که فیلتر متوسط گیر برای حذف نویز در تصاویر و همچنین آشکار نمودن جزئیات تصویر فیلتری ساده و مفید می باشد، ویژگی این فیلتر آن است که مثلاً اگر ابعاد آن ۳*۳ باشد ، زمانی که ۹ عدد پیکسل داخل فیلتر را با هم جمع کنیم مقدار آن ۱ می‌شود.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

$\frac{1}{49}$	$\frac{1}{49}$					

***مهم**

هر چه فیلتر بزرگتر باشد تصویر بهبود یافته خروجی مات‌تر شده و جزئیات ریز تصویر ناپدید می‌شود.

***مهم**

پس مهمترین نوع فیلتر، فیلتری با ابعاد کوچک است به دو دلیل یکی اینکه جزئیات ریز تصویر در آن از بین نمی رود نکته مهم دوم سرعت بالای پردازش در فیلترهای کوچک به دلیل تعداد ضربهای کم‌تر است.

فیلتر هموارسازی وزن دار: weighted smoothing filter

بیشتر فیلترهای هموار ساز موثر می توانند با داشتن وزنهای مختلف در پیکسلهای همسایه توسط تابع متوسط گیر تولید نشود وزن دار بودن به این معنی است که همه pixelها نباید هم وزن باشند و pixelهای نزدیک به مرکز وزن آنها بیشتر است به عبارتی، مهمتر هستند، اغلب آنها را فیلترهای متوسط گیر هم می نامند.

یه فیلتر هم وار ساز ۳*۳ وزن گیر اش به صورت زیر

$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$
$\frac{2}{16}$	$\frac{4}{16}$	$\frac{2}{16}$
$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$

عیب pixelهای از دست رفته در لبه ها (در لبه ها چه اتفاقی می افتد؟)

مثلاً مرکز ماسک را میلر ۳*۳ قسمتی به بیرون بیافتد.

برای روش دوم یعنی هر کدام می رود جایگزین قسمت مرزی خود می شود.

چه رویکردی برای مقابله با pixel های از دست رفته سه تصویر وجود دارد؟

یکی از روشها برای برطرف کردن این عیب pad the image است که در واقع با اضافه کردن صفر یا یک به جای pixel هایی که وجود ندارد این عیب را اصلاح می کند که در اصطلاح به آن zero padding نیز گفته می شود.

روش دوم برای رفع این عیب replicate boarder pixel می باشد. در این روش pixelهای مرزی به جای pixel هایی که وجود ندارد قرار می گیرند.

Correlation & convolution:

عملیات filtering که تاکنون از آن استفاده کردیم در حقیقت همان correlation است، عملیات convolution دقیقاً مشابه عملیات filtring بالا می باشد اما تنها یک تفاوت ضعیف دارد که filter فوق، 180 در جهت عقربه‌های ساعت می‌چرخد.

a	**b**	**c**
d	**e**	**f**
g	**h**	**i**

x	**y**	**z**
u	**v**	**m**
l	**m**	**n**

Lprocess=a*n+b*m+c*l+d*w+e*v+f*u+g*z+h*y+i*x

دستور imfilter (I,mask)

برنامه‌ای بنویسید که یک تصویر را از ورودی دریافت و توسط دستورات ماتریس در MATLAB دو فیلتر متوسط گیر ۹*۹ و ۵*۵ تولید کرده بر روی تصویر مورد نظر اعمال نموده و سپس تصاویر را در خروجی نمایش دهد.

```
I=imread ('f.jpg');
F1=ones (9,9)/81;
F2=ones (5,5)/25;
M1=imfilter(I,f1);
M2=imfilter (I,f2)
Figure(1),imshow(m1);
Figure(2),imshow(m2);
;
```

دستور fspecid،دستوری است که برای تولید mask متوسط گیر مورد استفاده قرار می گیرد و دارای دو ورودی است، ورودی اول تابع‌ای با نام overage است که مشخص می کند نوع mask متوسط گیر است و ورودی دوم آن بعد آن mask که باید عددی فرد باشد.

Fspecial('average',3)

دستور imnoise: این دستور هر نوع نویزی را بر روی تصویر ایجاد می کند، دارای سه ورودی است، ورودی اول ماتریس تصویر اصلی است ورودی دوم نوع nois موردنظر است و ورودی سوم غلظت نویز است که عددی بین صفر و یک است.

Imnoise (I,'salt spepper, ' غلظت)

دستور med filt 2، فیلتر medium یا میانه را ایجاد میکند.

*مثال: برنامه‌ای بنویسید که یک تصویر را از ورودی دریافت کرده سپس ۴ ماسک، فیلتر متوسط گیر با ابعاد ۳، ۹، ۱۵ و ۳۵ تولید کرده و هر ۴ ماسک را بر روی تصویر اعمال نموده و سپس تصویر حاصل را در خروجی نمایش دهد.

I=imread ('f.jpg');
F_3= fspecid ('average',3);
F_9= fspecid ('average',9);
F_{15}= fspecid ('average',15);
F_{35}= fspecid ('average',35);
M_3=imfilter (I,f_3)
M_9=imfilter (I,f_9)
M_{15}=imfilter (I,f_{15})
M_{35}=imfilter (I,f_{35})
Figure(1),imshow(m_3);
Figure(2),imshow(m_9);
Figure(3),imshow(m_{15});
Figure(4),imshow(m_{35});

برنامه‌ای که یک تصویر رنگی را دریافت کرده، nais فلفل نمکی با غلظ ۰/۱ را بر آن اعمال کرده و در خروجی نمایش دهد سپس با استفاده از یک mask مناسب مجدداً nais را از تصویر حذف کرده و در خروجی نمایش دهد؟

M=rgb 2 gray (I)

M_1= imnoise (m, 'salt and pepper',0.1)

Figure(1), imshow (m_1)

M_2=med filt2(m_1)

Figure(2),imshow (m_2)

نمونه سوالات

بسمه تعالی

۱. برنامه ایی بنویسید که تصویر 11. Jpg را از ورودی دریافت کرده، نویز فلفل نمکی به آن اعمال کرده و از یک بار از فیلتر حذف نویز و یک بار از فیلتر تطابقی برای رفع نویز آن استفاده نموده، در خروجی در یک پنجره تصویر نویزدار، تصویر خاکستری، و دو تصویر حذف نویز شده را نمایش دهد.

```
a=imread('11.jpg');
b=rgb2gray(a);
c=imnoise(b,'salt & pepper',0.5);
d=medfilt2(c);
e=wiener2(b,[3,3]);
subplot(2,2,1),imshow(c)
subplot(2,2,2),imshow(b)
subplot(2,2,3),imshow(d)
subplot(2,2,4),imshow(e)
```

۲. برنامه ایی بنویسید که تصویرx.bmp را از ورودی دریافت کرده، نویز لکه ایی با غلظت ۰,۸ به آن اعمال نموده ، در خروجی را نمایش دهد.

```
a=imread('x.bmp');
c=imnoise(a,'speckle',0.8);
imshow(c)
```

۳. برنامه ایی بنویسید که یک تصویر رنگی با نام 7.pngکه دارای نویز است را از ورودی دریافت کرده، با استفاده از فیلتر تطابقی ۳* ۳ رفع نویز نموده هر سه تصویر را در خروجی نمایش دهد.

```
f=imread('m7.png');
l=rgb2gray(f);
e=wiener2(l,[3,3]);
figure,imshow(e)
figure,imshow(f)
figure,imshow(l)
```

۴. برنامه ایی بویسید که چهار فیلتر میانگین با ماتریس مربعی به ابعاد ۳*۳ و ۹*۹ ، ۱۵*۱۵ و ۳۵*۳۵ طراحی کرده بر روی تصویری با نام fi.tif اعمال نموده و هر چهار تصویر پس از اعمال فیلتر را در خروجی نمایش دهد.

```
I=imread('fi.tif');
M3=fspecial('average',3);
M9=fspecial('average',9);
M15=fspecial('average',15);
M35=fspecial('average',35);
J3=imfilter(I,M3);
J9=imfilter(I,M9);
J15=imfilter(I,M15);
J35=imfilter(I,M35);
figure,imshow(J3)
figure,imshow(J9)
figure,imshow(J15)
figure,imshow(J35)
```

۵. برنامه ایی بویسید که یک فیلتر میانگین با ماتریس گرد با ابعاد ۳*۳ طراحی کرده بر روی تصویری با نام f.tif اعمال نموده و تصویر پس از اعمال فیلتر را در خروجی نمایش دهد.

```
I=imread('f.tif');
M3=fspecial('disk',3);
J3=imfilter(I,M3);
imshow(J3)
```

۶. برنامه ایی بنویسید که یک تصویر ۴ فریمی با نام f.tif را به فیلم تبدیل کند، و فیلم را در خروجی نمایش دهد.

```
for x=1:4
    w(:,:,:,x)=imread('f.tif',x);
end
filme=immovie(w);
implay(filme);
```

۷. برنامه ایی بنویسید که یک تصویر رنگی با نام r.jpg را دریافت کرده و با یک آستانه مناسب آن را به تصویر سیاه و سفید تبدیل کرده و در خروجی نمایش دهد. همچنین مقدار آستانه بدست آمده را نیز نشان دهد.

```
x=imread('r.jpg');
b=graythresh(x);
a=im2bw(x,b)
subplot(1,2,1),imshow(x)
subplot(1,2,2),imshow(a)
b
```

۸. برنامه ایی بنویسید که تصویری رنگی را به Index تبدیل کند (تعداد نمونه برداری از رنگ تصویر ۲۰۰ می باشد) و هر دو تصویر را در خروجی نمایش دهد.

```
p=imread('figa.jpg');
[x,map]=rgb2ind(p,200);
figure,imshow(p)
figure,imshow(x,map)
```

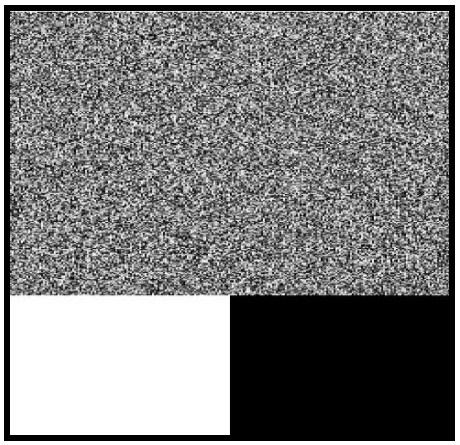
۹. برنامه ایی بنویسید که یک تصویر رنگی را دریافت کرده، ابتدا آن را خاکستری نموده و سپس تبدیل به سیاه و سفید کند در انتها هر سه تصویر را در خروجی نمایش دهد.

```
p=imread('figa.jpg');
q=rgb2gray(p);
y=im2bw(p);
figure,imshow(p)
figure,imshow(q)
figure,imshow(y)
```

۱۰. تصویر زیر را تولید کنید.

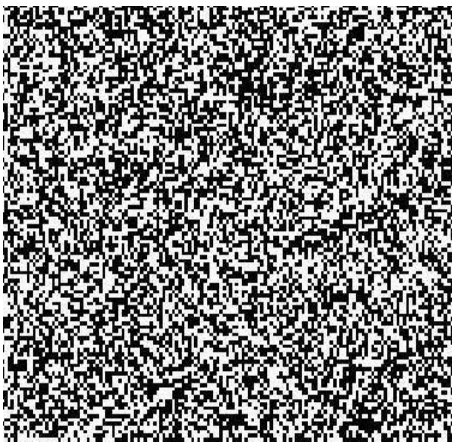
```
a=ones(120,120);
b=zeros(120,120);
c=[a b];
imshow(c)
```

۱۱. تصویر زیر را ایجاد کنید.



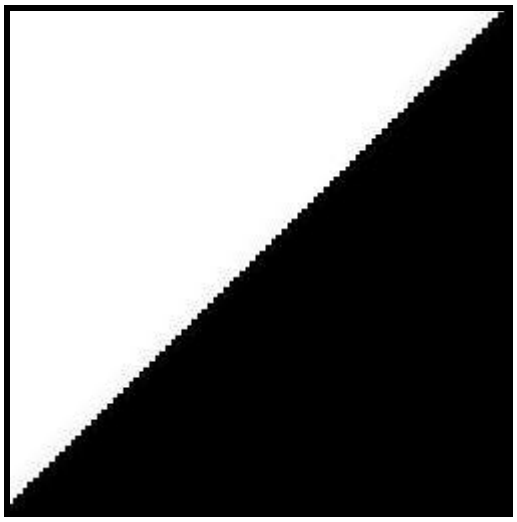
```
a=ones(120,120);
b=zeros(120,120);
c=rand(240,240);
d=[a b];
e=[c;d];
imshow(e)
```

۱۲. تصویر زیر را ایجاد کنید.



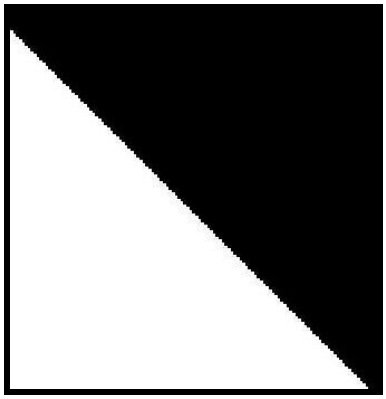
```
row=256;
col=256;
img1=rand(row,col);
img2=round(img1);
figure,imshow(img2)
```

۱۳. تصویر زیر را ایجاد کنید.



```
row=256;
col=256;
mat(1:row,1:col)=0;
for i=1:row;
    for j=1:col-i;
        mat(i,j)=1;
    end;
end
imshow(mat)
```

۱۴. تصویر زیر را ایجاد کنید.



```
row=256;
col=256;
mat(1:row,1:col)=0;
    for i=1:row;
        for j=1:i;
```

```
        mat(i,j)=1;
    end;
end
imshow(mat)
```

۱۵. برنامه ایی بنویسید که بخشی از یک تصویر را از مختصات (۱۰۰و۱۰۰) به طول ۲۰۰ و عرض ۲۰۰ جدا کرده به اندازه ۹۰ درجه خلاف جهت حرکت عقربه های ساعت چرخش داده هر سه تصویر را در خروجی نمایش دهد.

```
a=imread('pen.jpg');
b=imcrop(a,[100 100 200 200]);
c=imrotate(b,90);
subplot(1,3,1),imshow(a)
subplot(1,3,2),imshow(b)
subplot(1,3,3),imshow(c)
```

۱۶. برنامه ایی بنویسید که یک تصویر و هیستوگرام آن را با هم در خروجی نمایش دهد.

```
a=imread('pen.jpg');
b=rgb2gray(a);
subplot(1,2,1),imshow(b)
subplot(1,2,2),imhist(b)
```

۱۷. عملیات جمع و تفریق تصاویر با اعداد، Brightness تصویر را افزایش و کاهش می دهد و عملیاتی مانند ضرب، تقسیم و توان باعث افزایش و کاهش Contrast تصویر می شوند.

۱۸. برنامه ایی بنویسید که Contrast تصویر را ۴ برابر کند و هر دو تصویر را در خروجی نمایش دهد.

```
a=imread('pen.jpg');
c=immultiply(a,4);
  subplot(1,2,1),imshow(a)
  subplot(1,2,2),imshow(c)
```

۱۹. برنامه ایی بنویسید که Contrast تصویر را ۱/۳ کند و هر دو تصویر را در خروجی نمایش دهد.

```
a=imread('pen.jpg');
c=imdivide(a,3);
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(c)
```

۲۰. برنامه ایی بنویسید که Brightness تصویر را ۱۰۰ واحد اضافه کند.

```
a=imread('pen.jpg');
c=imadd(a,100);
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(c)
```

۲۱. برنامه ایی بنویسید که Brightness تصویر را ۱۰۰ واحد کم کند.

```
a=imread('pen.jpg');
c=imsubtract(a,100);
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(c)
```

۲۲. تصویری به صورت هوایی از یک زمین کشاورزی گرفته شده است و نیاز به مشخص شدن تغییر سطح از روی رنگ داریم، نام فیلتر مورد نظر را بیان کرده و با کمک چهار رنگ مرز پستی و بلندی تصویر را مشخص کنید و هر دو تصویر را در خروجی نمایش دهید.

فیلتر حد فاصل

```
a=imread('pen.jpg');
c=rgb2gray(a);
subplot(1,2,1),imshow(c)
subplot(1,2,2),imcontour(c,4)
```

۲۳. می خواهیم لبه های موجود در تصویر pen.jpg برایمان نمایان شود، از چه الگوریتمی استفاده می شود و برنامه آن را بنویسید.

prewittالگوریتم

```
a=imread('pen.jpg');
f=fspecial('prewitt');
x=imfilter(a,f)
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(x)
```

۲۴. یک فیلتر حرکت دهنده طراحی کنید که حرکت تصویر به سمت بالا و پایین بوده وشدت حرکت به اندازه ۵۰ باشد.

```
a=imread('pen.jpg');
f=fspecial('motion',50,90);
x=imfilter(a,f);
```

```
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(x)
```

۲۵. یک فیلتر حرکت دهنده طراحی کنید که حرکت تصویر به سمت چپ و راست بوده وشدت حرکت به اندازه ۴۰ باشد.

```
a=imread('pen.jpg');
f=fspecial('motion',40,0);
x=imfilter(a,f);
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(x)
```

۲۶. با کمک یک فیلتر قالب تصویر خود را نمایش دهید، نام این فیلتر و اندازه آن را نیز بنویسید. (مقدار آلفا ۰,۵ است).

فیلتر لاپلاس اندازه غیر قابل تغییر ۳*۳

```
a=imread('pen.jpg');
f=fspecial('laplacian',0.5);
x=imfilter(a,f);
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(x)
```

۲۷. می خواهیم بر طبق هیستوگرام کیفیت شدت نور تصویر را بهینه کنیم، تصویر اصلی، بهینه شده و هیستوگرام را در خروجی نمایش دهید.

```
a=imread('pen.jpg');
b=rgb2gray(a);
c=histeq(b);
subplot(1,3,1),imshow(b)
subplot(1,3,2),imshow(c)
subplot(1,3,3),imhist(c)
```

۲۸. تصویر را از ورودی گرفته سایز آن را ۶۰٪ کوچک کرده هر دو تصویر را در خروجی نمایش دهید.

```
a=imread('pen.jpg');
b=imresize(a,0.6);
subplot(1,2,1),imshow(a)
subplot(1,2,2),imshow(b)
```

۲۹. تصویر قسمت الف را داریم، این تصویر رنگی است، حال می خواهیم تصویر قسمت ب را ساخته و به شکل زیر دو تصویر را ادغام کرده در خروجی نمایش دهیم.

ب الف

```
a=imread('m6.jpg');
b=im2bw(a);
c=ones(size(b));
d=[b,c];
figure,imshow(d)
```

۳۰. برنامه ایی بنویسید که بخشی از یک تصویر را از مختصات (۴۰و۴۰) به طول ۲۰۰ و عرض ۲۰۰ جدا کرده به اندازه ۲۰ درجه در جهت حرکت عقربه های ساعت چرخش داده هر سه تصویر را در خروجی نمایش دهد.

```
a=imread('pen.jpg');
b=imcrop(a,[40 40 200 200]);
c=imrotate(b,-20);
subplot(1,3,1),imshow(a)
subplot(1,3,2),imshow(b)
subplot(1,3,3),imshow(c)
```